WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, March 05, 2004

Hide?	Set Name	Query	Hit Count
	DB=USPT; PLUR=NO; OP=OR		
☐	L53	(l50 or l51 or L52) and ((clean-up or (clean adj1 up) or cleanup) same (folder$ or file$))	1
☐	L52	whitney-david-c.in.	3
☐	L51	sherman-roman.in.	7
☐	L50	mansour-peter-m.in.	3
☐	L49	mansour-peter.in.	0
☐	L48	L44 and ((record$ or file$) same flag$)	73
☐	L47	L43 and (offline or clean).clm.	3
☐	L46	L43 and (offline or clean).ab.	6
☐	L45	L43 and (offline or clean).ti.	2
☐	L44	L43 and (offline or clean)	118
☐	L43	L4 and ((707/$)!.CCLS.)	696
☐	L42	L40 and (offline or clean)	62
☐	L41	L40 and (offline or clean)	62
☐	L40	L31 and L39	290
☐	L39	L28 and ((file$ or folder$) same flag$)	1561
☐	L38	L32 and (offline or clean)	42
☐	L37	L31 and (offline or clean).ab.	39
☐	L36	L31 and (offline or clean).ti.	7
☐	L35	L30 and (offline or clean).ab.	1
☐	L34	L30 and (offline or clean).ti.	0
☐	L33	L30 and (offline or clean)	191
☐	L32	L30 and L31	203
☐	L31	(((707/$)!.CCLS.))	12330
☐	L30	L29 and (records same flag$)	915
☐	L29	((delet$ or discard$ or remov$) same flag$)	11838
☐	L28	((delet$ or discard$ or remov$) same flag)	9778
☐	L27	L26 and (cleanup or (clean adj1 up) or clean-up)	52
☐	L26	L25 and ((flag or flags) same (folder$ or file$))	183
☐	L25	L24 and ((file$ or folder$) same message$)	577

09/409,630

h e b b cg b chh e f b f c e c e

☐	L24	L23 and ((delet$ or discard$ or remov$) same message$)	729
☐	L23	(hierarch$ same (file$ or folder$))	5541
☐	L22	L21 and (cleanup or (clean adj1 up) or clean-up)	2
☐	L21	L20 and ((flag or flags) same (folder$ or file$))	14
☐	L20	L19 and ((file$ or folder$) same message$)	51
☐	L19	L18 and ((delet$ or discard$ or remov$) same message$)	99
☐	L18	(((707/100)!.CCLS.))	1405
☐	L17	L16 and (cleanup or (clean adj1 up) or clean-up)	6
☐	L16	L15 and ((flag or flags) same (folder$ or file$))	53
☐	L15	L13 and ((file$ or folder$) same message$)	245
☐	L14	L13 and ((delet$ or discard$ or remov$) same message$)	141
☐	L13	(((707/1)!.CCLS.))	1495
☐	L12	L10 and ((flag or flags) same (folder$ or file$))	22
☐	L11	L10 and (cleanup or (clean adj1 up) or clean-up)	9
☐	L10	L9 and ((file$ or folder$) same message$)	226
☐	L9	((delet$ or discard$ or remov$) same message$).ab.	976
☐	L8	((delet$ or discard$ or remov$) same message$).ti.	24
☐	L7	((delet$ or discard$ or remov$) same message$)	20790
☐	L6	L5 and (cleanup or (clean adj1 up) or clean-up)	144
☐	L5	L2 and ((discard$ or delet$ or remov$) same flag)	812
☐	L4	((discard$ or delet$ or remov$) same flag)	9778
☐	L3	L2 and (flag same message$)	1457
☐	L2	L1 and (message$ same (file$ or folder$))	5659
☐	L1	((discard$ or delet$ or remov$) same message$)	20790

END OF SEARCH HISTORY

h e b b cg b chh e f b f c e c e

☐ Generate Collection

L3: Entry 7 of 30 File: USPT Jul 10, 2001

DOCUMENT-IDENTIFIER: US 6260049 B1
TITLE: Automated shelf management system and process for tracking and purging file folders in a file storage facility

US Patent No. (1):
6260049

Detailed Description Text (64):
In operation, following the flowcharts in FIGS. 12a-12e, the log-out subroutine begins with step 210, in which the program is waiting for a folder. The user places a folder 52 on the electronic scale 130 and scans the bar code label 82 with the bar code reader 131 or alternatively enters a folder number and volume number via the keyboard 92. In steps 214 and 216, the program determines if the folder was scanned; if not scanned, the system sets a flag. In step 216, as an option, a determination is made whether the file folder is the last volume, by scanning the volume number in step 218. If there is no volume number, the last volume flag is set in step 220, indicating that the file folder has only one volume. If there is volume number, the last volume flag is reset in step 222. The volume number is scanned in step 224 and the volume number is stored in step 226.

Detailed Description Text (72):
The system next proceeds to print a bar code label if needed. First, in step 288, a determination is made whether the bar code label printer 132 is in use, or ready to print a new label. In step 290, if is determined whether the bar code for the file folder 52 had been scanned. If the bar code was not scanned, the assumption is made that the file folder 52 requires bar code label to be printed, and the label is printed in step 292, and the non-scanned flag is reset in step 293.

Detailed Description Text (74):
If there are pending requests, as determined in step 294, the file folder 52 is logged out. The program first determines in step 302 if there is more than one pending request; if there is, the operator selects the next logout request in step 304. After step 304, the program determines if the files are locked in step 306. If the files are not locked, an error message is displayed in step 308 and the programs again exits at step 300. If all the files are locked in step 306, the program proceeds in step 310 to determine whether the scale 130 is in use, and if the weight on the scale is valid in step 312. If the weight is not valid, an error message is displayed in step 314 and the program exits in step 316. If the weight is valid, the size of the folder is calculated in step 318. If the scale was not is use, or after completion of calculating the file size, the master file is updated in step 322. In step 324, a determination is made if the present file folder 52 needs a new folder. If a new folder is required, the system user is alerted in step 326, so that the folder data may be properly supplied. Otherwise, a routing slip is printed to accompany the file to the requestor's location. If the user manually inputs the folder number into the system via the keyboard, the system prints a new bar code label in step 332. In step 334 the scanned/volume flags are reset and the program exits in step 336 to await a new folder transaction.

Detailed Description Text (118):
Referring now to FIG. 32, the Measure Shelves flowchart is shown. The measure shelves subroutine allows the user to record, measure, and label folders on each of the shelves. In step 576, the measure shelves SHLWS is initiated, causing the scan prompt to be displayed in step 578. The folder number is input in step 582. If the file folder was not scanned in step 582, indicating that there is no label, a label will be printed in step 584. If the file has a volume number in step 586, the volume number will be stored in step 588. If there is no volume number, the file will be

considered to be a single file folder. In step 590, the user indicates if the file folder is in-file or out-of-file. If the file is out, the "folder out" flag is set in step 592. If the folder is in, the file will be updated in step 594 and the user operator is prompted to scan the folder.

☐ Generate Collection

L12: Entry 2 of 15 File: USPT Jun 4, 2002

DOCUMENT-IDENTIFIER: US 6401112 B1
TITLE: Method and apparatus for synchronizing an Email client on a portable computer system with an Email client on a desktop computer

Abstract Text (1):
A fully integrated email system for a desktop computer with an associated palmtop computer is disclosed. The portable computer has an email client for viewing incoming email messages and composing outgoing email messages. The personal computer has an email synchronization conduit that synchronizes email on the portable computer email client with email for the desktop computer system. The email synchronization conduit ensures that the email state on the portable computer system matches the email state on the desktop personal computer system exactly. Thus, if an email message is deleted on the portable computer system then that email message will be deleted from the desktop personal computer system. Similarly, if an email message is deleted on the desktop personal computer system then that email message will be deleted from the portable computer system.

Detailed Description Text (12):
The portable computer system requires a connection to a computer network infrastructure to receive new email and to send newly composed email messages. To connect with the computer network infrastructure, this document describes a synchronization environment wherein the portable computer is coupled to a personal computer using a serial link as depicted in FIGS. 1a, 1b, and 2a. However, many other methods of connecting the portable computer system to a personal computer system as described in the patent application "Method And Apparatus Por Synchronizing A Portable Computer System With A Desktop Computer System" filed on Jan. 30, 1997 with Ser. No. 08/792,166, now U.S. Pat. No. 6,006,274.

Detailed Description Text (21):
At step 335, the email conduit 233 checks to see if this is a synchronization with the same personal computer that was used to perform the previous synchronization. If this is the same personal computer, then the email from the portable computer system will have valid status flags that specify the new, changed and deleted email from the portable computer system. However, if this is not the same personal computer that was used in the previous synchronization then the conduit proceeds to step 337 where it compares the email from the portable computer system with the email from the last synchronization to determine the new, changed and deleted email from the portable computer system. Additional information about the system of using a state from a previous synchronization is available in U.S. Pat. No. 5,727,202 entitled "Method and Apparatus for Synchronizing Multiple Files On Two Different Computer Systems" with Ser. No. 08/544,927, filed on Oct. 18, 1995, now U.S. Pat. No. 5,727,202.

End of Result Set

☐ Generate Collection

L35: Entry 1 of 1 File: USPT Aug 29, 1995

DOCUMENT-IDENTIFIER: US 5446901 A
TITLE: Fault tolerant distributed garbage collection system and method for collecting network objects

Abstract Text (1):
A distributed computer system includes a multiplicity of concurrently active processes. Each object is owned by one process. Objects are accessible to processes other than the object's owner. Each process, when it receives a handle to an object owned by any other process, sends a first "dirty" message to the object's owner indicating that the object is in use. When a process permanently ceases use of an object handle, it sends a second "clean" message to the object's owner indicating that the object is no longer in use. Each object's owner receives the first and second messages concerning usage of that object, stores data for keeping track of which other processes have a handle to that object and sends acknowledgement messages in return. The receiver of an object handle does not use the handle until its first message is acknowledged. Periodically, the object's owner sends status request messages to other processes with outstanding handles to that object to determine if any of those processes have terminated and updates its stored object usage data accordingly. A garbage collection process collects objects for which the usage data indicates that no process has a handle. The first and second messages include sequence numbers, wherein the sequence numbers sent by any process change in value monotonically in accordance with when the message is sent. Object owners ignore any message whose sequence number indicates that it was sent earlier than another message for the same object that previously received from the same process.

Detailed Description Text (32):
Referring to FIG. 2, in accordance with the present invention a shared object cannot be deleted so long as its usage table 174 contains even a single entry having an "in use" (or "dirty") flag that is still set. Unfortunately, processes that terminate, normally or abnormally, cannot be expected to reliably notify the owners of all network objects for which those processes have object handles. In other words, the local garbage collection procedure 180 and cleanup demon 186 of the terminated process may not be able to generate and transmit "clean messages" for all surrogate objects in the process before the process terminates. Therefore, in the present invention, each shared object's owner is provided with a facility for determining which processes that had handles to the shared object have terminated.

Detailed Description Text (39):
When an object owner receives a "clean" message from a client process, it compares the sequence number of the received message with the sequence number stored in the corresponding object's usage table for the process that sent the clean message. If the sequence number indicates that the clean message was created later than any other message concerning the same network object that has been received from that process, the entry for that process in the object usage table is updated by storing a "not in use" flag (also herein called a "clean" flag) value and replacing the stored sequence number with the sequence number received with the clean message. As long other records in the usage table 174 indicate that at least one handle to the shared object is still held by a client process, the usage table entry for the process that has released the object handle is not deleted. Rather the entry is retained so that the most recent sequence number for messages from the corresponding process will be retained (e.g., to protect against "dirty messages" whose transmission was delayed by a communication failure).

Detailed Description Text (41):
When the shared object garbage collector 181 receives a "process failed" message

from the failed process detection procedure 194; which identifies the process that has failed or is presumed to have failed, the shared object garbage collection procedure 181 performs the following steps: (1) it locates each usage table 174 that contains an entry for the identified failed process, and then deletes that entry; and (2) if the usage table 174 of an object has no remaining records, or if the none of the remaining records have an "in use" flag which is set, it deletes that object's usage table 174 and deletes the object's entry in the object table 162. Thus, a "process failed" message is treated like a global "clean" message directed to all shared objects identified in the object table 162.

☐ Generate Collection

L38: Entry 4 of 36 File: USPT Jan 7, 2003

DOCUMENT-IDENTIFIER: US 6505214 B1
TITLE: Selective information synchronization based on implicit user designation

Detailed Description Text (19):
In a preferred embodiment, the information (e.g., subfolders of an expanded folder) get downloaded to the device when a user expands the folder, but the information is not synchronized at that time. Instead, the subfolders of interest are marked for synchronization and will be synchronized on subsequent connections. The difference between downloading the subfolders and synchronizing them is that any changes previously made on the companion device to the subfolder list in question do not get propagated to the server during the download operation, but those changes are propagated to the server during synchronization. As an example, during an offline session, a user may create a subfolder of a folder that has not been previously expanded. Next, the user connects to the server and expands that folder causing subfolders of this folder existing on the server to be downloaded to the device. However, the subfolder that was created on the device won't be propagated to the server until the next connection procedure.

Detailed Description Text (24):
Folders that are synchronized are thereby made available on the H/PC. The synchronization process also makes sure that any changes made to that folder list on either the device or the server are reconciled. As an example, a creation or deletion of a folder on the device while the device was offline would be propagated to the server during synchronization. In the same way, creation or deletion of folder on the server would be propagated to the device during synchronization. Preferably, the synchronization process also synchronizes any folder rename operations. Moreover, other folder operations may be reconciled during the folder list synchronization.

Detailed Description Text (28):
A mail store is provided in the H/PC, and includes two separate databases: a message database, and a folder database. These two databases house the data records representing both the message and folder data within the user-defined, hierarchical folder architecture. In one embodiment of the invention, the expanded flag represents a field of the folder database in each database record. This field is then monitored in each record to determine the state of the expanded flag associated with the corresponding folder. Testing of this folder database field is represented by decision operations 252 and 258 of FIG. 6. As more folders are synchronized, additional records are accordingly generated in the folder database. As will be appreciated by those skilled in the art, other manners of storing the expanded flag other than via a database field are possible, including mapped memory structures or other addressable storage structures whereby stored expanded flag indicators can be located.

Detailed Description Text (44):
The synchronization operations of FIG. 10 are carried out on the client. All subfolders in the client list are marked by mark operation 450 with an identifying flag, hereinafter referred to as the "delete flag." The delete flag provides an indication as to what subfolders were present in the device list at the outset of synchronization. As will become more apparent from the ensuing description, this flag is used to ultimately direct the client to delete a particular subfolder that has been deleted from the server since the last connection.

Detailed Description Text (45):
A server list synchronization loop is then performed. Retrieving operation 452 retrieves the server list from the server, wherein the server list is a list of

subfolders for a particular folder as this operation is conducted on a folder to folder basis. In one embodiment, the top-level folder (e.g., Service Hierarchy Folder 200 of FIG. 5) does not exist on the server and only exists on the client, and therefore will not be "retrieved" from the server list. The subfolders associated with that particular service will exist on the server, and will be downloaded as part of the server list. A first subfolder of the server list is recognized by operation 454, and operation 456 determines whether that subfolder is in the client list. If so, operation 458 clears the delete flag in the client list. To "clear" the delete flag simply means to "unset" the delete flag that was initially set, and does not suggest or require any particular binary value or code. Any predetermined values may be used as to identify the "set" or "clear" conditions. Essentially, clearing the delete flag indicates that the server list subfolder is present on the client, and should not be deleted from the client because it is present on the server. Once the delete flag has been cleared in the client list, operation 466 detects whether there are more subfolders in the server list to consider, and if so, operation flow returns to decision operation 456 where the next subfolder in the device list is considered.

Detailed Description Text (47):
If operation 456 detects that the particular subfolder of the server list is not present in the client list, operation 460 determines whether that subfolder was deleted from the client while offline (i.e., not connected to the server) since the last connection. If so, that subfolder is deleted from the server list by operation 462, as deletion of the subfolder from the device indicates the user's desire to have that subfolder removed from the hierarchy. Once deleted, operation 466 detects whether there are more subfolders in the server list to consider, and if so, operation flow returns to decision operation 456 where the next subfolder in the client list is considered.

Detailed Description Text (48):
The fact that a device subfolder has been deleted while the device was offline is preferably indicated by a special "need to delete" flag. When the user deletes a folder while offline, it is not necessarily deleted at that time and is instead marked with the "need to delete" flag. Then, when constructing the client subfolder list for the parent of this folder, the folder is included into the list in order to match it up to a corresponding server folder. Following the match process, the folder is deleted along with the corresponding server folder during the next online session.

Detailed Description Text (49):
If the subfolder was not deleted from the client while offline, create operation 464 creates the subfolder on the client. A subfolder is created on the device at operation 464 because the subfolder is known to be present in the server list (i.e., operation 454 or 466), it is not present on the client (i.e., operation 456), and it was not deleted from the client since the last connection (i.e., operation 460). Therefore, to synchronize subfolders in the server and client lists, a corresponding subfolder must be created in the client list. Once created, operation 466 detects whether there are more subfolders in the server list to consider, and if so, processing returns to decision operation 456 where the next subfolder in the client list is considered.

Detailed Description Text (50):
A client list synchronization loop is then performed. A first subfolder of the client list is considered at operation 468, and operation 470 determines whether that subfolder was created on the client while offline since the last connection. When subfolders are created on the client while offline, they are marked with a special creation flag. This creation flag is checked at decision operation 470. If the creation flag indicates that the subfolder was created on the client while offline, operation 472 creates a corresponding subfolder on the server, after which the creation flag is cleared. Then, operation 478 determines whether there are more subfolders in the client list to consider. If so, processing returns to decision operation 470 where the next subfolder in the client list, if any, is considered.

Detailed Description Text (51):
If operation 470 determines that the subfolder was not created on the client while

offline since the last connection, test operation 474 tests whether the "delete flag" is still set. If so, it means that subfolder was not previously found in the server list, and therefore needs to be deleted from the client, which is accomplished by operation 476. Operation 478 determines whether there are more subfolders in the client list to consider, and if so, processing returns to decision operation 470 where the next subfolder in the device list, if any, is considered.

Detailed Description Text (52):
The embodiments of the invention as presently described present techniques for synchronizing subsets of information between two systems, based on implicit perception of a user's need to synchronize such information and without the need for explicit designation by the user. The subsets of information, such as subfolders in a folder hierarchy, are ascertained by the invention through normal operations performed by the user rather than explicit designation. The information subsets are marked (in one embodiment, with expansion flags set on e-mail folders) to indicate that the particular information is to be synchronized. However, it may also be desirable to provide a manner of excluding from the synchronization process particular information previously marked for synchronization (in one embodiment, this corresponds to clearing or "unsetting" the expansion flags on certain folders). The description in connection with FIG. 11 therefore describes one embodiment of logical operations for removing folders/subfolders from synchronization.

Current US Original Classification (1):
707/201

Current US Cross Reference Classification (1):
707/1

Current US Cross Reference Classification (2):
707/200

Current US Cross Reference Classification (3):
707/203

Current US Cross Reference Classification (4):
707/8

Generate Collection

L38: Entry 9 of 36 File: USPT Apr 10, 2001

DOCUMENT-IDENTIFIER: US 6216127 B1
TITLE: Method and apparatus for processing electronic mail in parallel

Brief Summary Text (14):
A process can be used to perform multiple tasks or activities. Each process can be configured to perform one or more of these activities. Further, processes can be configured to run during a certain time period. Thus, for example, multiple processes can be configured to perform garbage collection. A garbage collector process can be further configured to, for example, clean up mail messages or scheduler messages, or clean up replication or directory registration information. Further, a garbage collector can be run at night to perform garbage collection on mail messages. Another garbage collector can be run during the daytime to perform garbage collection tasks.

Detailed Description Text (15):
Multiple processes, for example, can be configured to perform garbage collection. A garbage collector process can be further configured to, for example, clean up mail or scheduler messages, or clean up replication or directory registration information. One of the garbage collector processes can run at night to perform garbage collection on mail messages. Another garbage collector can be run during the daytime to perform garbage collection tasks.

Detailed Description Text (21):
The flags field for an instance of the postman process, for example, can be used to indicate that the postman instance perform local delivery, remote delivery, gateway processing, or notification. To illustrate further, the flags field for an instance of a garbage collector process can be used to indicate that the process cleanup registration records, or perform scheduler, directory, or mail garbage collection.

Detailed Description Text (87):
FIG. 11 illustrates a performServerAction process flow for a Postman process. At decision block 1102 (i.e., "remote flag set?"), if the remote flag is set, the Postman performs remote instance deletion at processing block 1104 and processing continues at decision block 1106. If not, processing continues at decision block 1106. At decision block 1106 (i.e., "gateway flag set?"), if the gateway flag is set, the Postman performs gateway processing at processing block 1108 and processing continues at decision block 1110. If not, processing continues at decision block 1110.

Current US Original Classification (1):
707/10

Current US Cross Reference Classification (1):
707/103R

Generate Collection

L48: Entry 43 of 67 File: USPT Sep 15, 1998

DOCUMENT-IDENTIFIER: US 5809527 A
TITLE: Outboard file cache system

Drawing Description Text (117):
FIGS. 120A, 120B, 120C, and 120D contain a flowchart of FLAGS processing which tests the flags in the File Descriptor when a segment hit occurs;

Drawing Description Text (138):
FIG. 141 contains a flowchart of NEW-BIT processing which tests whether the NEW flag in a File Descriptor should be set for the segment in process;

Detailed Description Text (44):
The file cache system permits Command Packets to be chained together. That is, a first Command Packet 452 may point to a second Command Packet, and the second Command Packet may point to a third Command Packet, and so on. The NEXT.sub.-- COMMAND.sub.-- PACKET 452d is used for chaining the Command Packets together. It contains the address of the next Command Packet in the command chain. If the CCF 452e (Command Chain Flag) is set, then NEXT.sub.-- COMMAND.sub.-- PACKET contains the address of the next Command Packet in the command chain. A chain of commands is also referred to as a "program." If CCF is clear, then no Command Packets follow the Command Packet in the command chain. The CCF is stored at Bit 5 of Word 3 in the Command Packet.

Detailed Description Text (104):
A cache indicator flag in the file descriptor table is used to identify when a file is cached by the File Cache System. If the cache indicator flag is set, Decision Step 606 forces Control Path 608 which leads to Step 610. Step 610 passes the I/O request parameters and control to the File Cache Handler Software 208 for further processing. If the cache indicator flag is not set, Decision Step 606 forces Control Path 612 to Decision Step 614. Decision Step 614 check whether the I/O request parameters specify that the file should be cached. If Decision Step 614 is positive, then Control Path 616 is followed to Step 618 where the cache indicator flag in the file descriptor table is set. Processing then proceeds to Step 610 which was discussed above. If the I/O request parameters do not indicate that a file should be cached, then Control Path 620 is followed to Step 622. Step 622 performs the necessary I/O processing for files which are not cached.

Detailed Description Text (111):
FIG. 24 shows a flow chart of the general processing for detecting when the processing of a Command Packet (or a chain) is complete. A Global Completion Flag and a Local Completion Flag are set by the Data Mover 110 after a Program Status Packet is written to Host Main Storage 16. A single Local Completion Flag is associated with each Program Initiation Queue and Status Packet Queue. When the File Cache Handler Software 208 detects that the Global Completion Flag is set, the Local Completion Flags are tested. If any of the Local Completion Flags are set, then the first Program Status Packet in the associated Status Packet Queue is retrieved and the status processed. The completion flags are continuously monitored for status processing.

Detailed Description Text (112):
Decision Step 702 checks whether the Global Completion Flag is set. Until the Global Completion Flag is set, no processing of Outboard File Cache status information is performed. After the Global Completion Flag has been set, processing proceeds to Step 704 where the Global Completion Flag is cleared. This allows the Data Mover to set the Global Completion Flag for the next Program Status Packet it returns. Step 706 gets the first Local Completion Flag.

Detailed Description Text (192):
Processing begins with Decision Step 902 checking whether there are any Destage Request Packets to process. This is accomplished by testing the DESTAGE.sub.-- REQUEST.sub.-- PACKETS flag 4601 in the Status Packet. If it is set, processing proceeds to Step 904. At Step 904, the DESTAGE.sub.-- REQUEST.sub.-- PACKET.sub.-- COUNT 894b in the Status Packet is used to determine the number of Destage Request Packets 896. For each of the Destage Request Packets in the Status Packet, an entry is made in an input queue for the Destage Process. The Destage Process recognizes the PRIORITY.sub.-- DESTAGE flag and gives the destage request priority. The Destage Process manages destaging segments referenced by the queued Destage Request Packets from Outboard File Cache 102 to Disks 106.

Detailed Description Text (207):
For each of the Segment State Packets 2110 returned in response to the RETURN SEGMENT STATE command, the Disabled Flag (DF) is tested. For each of the segments whose Disabled Flag is set and whose Segment Valid Flag (SVF) is cleared, a bad spot is identified in the file control table of the associated file, as indicated by Step 1142. Those segments whose Segment Valid Flag is set do not need to be marked as bad because the corresponding segment on the Disk is valid. A subsequent reference to a segment marked as bad will not be honored and an error status will be returned to the routine referencing the bad segment.

Detailed Description Text (237):
After the Outboard File Cache 102 has transferred the identified segments to Host Main Storage 16, each segment is checked as to whether it is valid. A segment is valid if it has not been written since the last time it was stored to disk. Otherwise the segment is invalid. The validity of a segment is indicated by the Segment Not Valid (SNV) flag in the Segment Information Packet returned in the DESTAGE Status Packet. Decision Step 1512 checks whether a segment is valid. If it is not, then control Path 1512n is followed to Step 1514 to handle the case where a segment is not valid.

Detailed Description Text (240):
Decision Step 1520 checks whether the data was successfully written to Disk 106. During the normal course of processing it is expected that the write to disk would be successful. For successful writes, control Path 1520y is followed to Step 1522 for reporting the status of the destage operation back to the Outboard File Cache 102. Step 1522 provides the File Cache Interface processing with the parameters required for a DESTAGE COMPLETE command. The Outboard File Cache may then clear the written flags and destage flags in the File Descriptor 508 to indicate that destaging is not longer required. After the Status Packet associated with the DESTAGE COMPLETE Command Packet is returned to the Destage Process, control Path 1522p is followed to Step 1524 where the temporary storage used for destaging is deallocated. Control then returns to decision Step 1504 to monitor the input queue for more destage requests.

Detailed Description Text (242):
For duplexed files the lost segments are recovered. Control from decision Step 1526 is passed to Step 1532 via control Path 1526y. Step 1532 invokes File Cache Interface processing with the parameters necessary for a DESTAGE COMPLETE command. In particular, those segments not destaged are identified with the Not Destaged (ND) flag in the DESTAGE COMPLETE Command Packet and are marked as written. The following steps then recover the data by using the duplicate leg of storage.

Detailed Description Text (294):
The DESTAGE AND PURGE DISK command is used to destage all data associated with a Disk 106 that resides in the Outboard File Cache 102 but does not reside on the Disk. This command may also be used to purge all data in the Outboard File Cache that is associated with a particular Disk. The Outboard File Cache 102 searches its File Space 502 for segments associated with the Disk identified in the Command Packet. For each segment found, if the purge flag in the Command Packet is set, and the segment has been written, it is destaged and placed in a purge pending state, otherwise it is purged. If the purge flag in the Command Packet is not set and the segment has been written, it is destaged and placed in a destage pending state,

otherwise no further processing is performed on the segment.

Detailed Description Text (315):
The DESTAGE AND PURGE FILES BY ATTRIBUTES command is used to destage and optionally purge segments of files that share one or more attributes, whereby the segments belonging to these files which reside in cache and do not reside on Disk 106 are destaged. The Outboard File Cache 102 searches File Space 502 for segments whose FILE.sub.-- ID matches the attributes specified in the Command Packet. The possible attributes include: all files local to a selected Host, all shared files, all temporary files, all catalogued files, and all files belonging to a selected application program. If the purge flag in the Command Packet is set and the segment has been written, it is destaged and placed in a PURGE PENDING stage, otherwise no processing is performed on the segment. If the purge flag in the Command Packet is not set and the segment has been written, it is destaged and place in a DESTAGE PENDING state, otherwise no processing is performed on the segment. FIG. 42 can be referenced as to how the DESTAGE AND PURGE FILES BY ATTRIBUTES command may be used.

Detailed Description Text (438):
A Recently Used Zone is used in the round-robin cache replacement scheme described herein. It is desirable to not reassign a segment which has been recently referenced because that segment is likely to be referenced again. A stage operation is saved if the segment which was not reassigned is referenced subsequent to it having been considered for reassignment The RECENTLY USED ZONE, along with the NEW flag in the File Descriptor 508, is used to mark segments which have been referenced recently. If a segment is referenced and it is within the Recently Used Zone, the NEW flag in the referenced segment's File Descriptor is set When the REPLACEMENT CANDIDATE eventually points to the referenced segment, the referenced segment win not be reassigned because it was recently referenced. The NEW flag will be cleared when the CURRENT REPLACEMENT CANDIDATE is advanced. The boundaries of the Recently Used Zone are defined by the REPLACEMENT CANDIDATE and the RECENTLY USED ZONE pointer which is 3n/4+i segments beyond the REPLACEMENT CANDIDATE pointer.

Detailed Description Text (461):
FIGS. 101A, 101B, 101C, and 101D contain a flowchart of the READ-WRITE routine. The READ-WRITE routine processes READ and WRITE commands sent from the Host 10. The READ-WRITE routine performs the set-up operations required for the Host Interface Adapter 214 to transfer data between the Host 10 and the Non-volatile Storage 220. Processing begins with calling the SEARCH routine at Step 6122. The SEARCH routine checks whether the segment referenced in the Command Packet is present in File Space 502. If it is, the HIT.sub.-- FLAG is set. After the SEARCH routine returns, processing proceeds to decision Step 6124.

Detailed Description Text (462):
If the HIT.sub.-- FLAG was set, then decision Step 6124 directs control to Step 6126 where the NEW-BIT routine is invoked. This routine checks whether the segment being referenced falls between the RECENTLY USED ZONE and the REPLACEMENT CANDIDATE discussed above. If it is behind the RECENTLY USED ZONE and ahead of the REPLACEMENT CANDIDATE, then the NEW.sub.-- BIT in the File Descriptor 508 is set. Setting the NEW bit removes the segment from consideration for reassignment on this cycle of the round robin processing.

Detailed Description Text (463):
After NEW-BIT processing, the processing proceeds to decision Step 6132. Decision Step 6132 checks whether any of the Group-1 flags are set. The Group-1 flags are set. The Group-1 flags are stored in the File Descriptor 508 and include SEGMENT.sub.-- BUSY, STAGE.sub.-- PENDING, DESTAGE.sub.-- PENDING, PURGE.sub.-- PENDING, TOTAL.sub.-- SEGMENT.sub.-- VALID, SPECULATIVE, STICKY.sub.-- COUNTER, and SEGMENT.sub.-- DISABLED. If any of the Group-1 flags are set, then processing proceeds to the FLAGS routine as shown by Step 6134 and then to decision Step 6136. Otherwise, processing proceeds directly to decision Step 6136.

Detailed Description Text (464):
The FLAGS routine exits the main line processing because some flag in the File Descriptor 508 is set which will not allow normal hit processing to proceed. For example, the segment may be in a STAGE PENDING state, in which case the Resend

RECOMMENDED.sub.-- ACTION is returned to the Host 10. The FLAGS processing will be considered in greater detail in its accompanying flowchart.

Detailed Description Text (465):
Decision Step 6136 checks whether the PRIOR.sub.-- MISS flag is set. The PRIOR.sub.-- MISS flag is set when one of the earlier segments referenced by the command resulted in a miss. When the PRIOR.sub.-- MISS flag is set, the READ-WRITE processing is forced in to a processing mode wherein the SEGMENT.sub.-- MISS.sub.-- TEMPLATE continues to be developed. No file data is transferred back to the Host 10 where some of the segments referenced by the command are not present in File Space 502, therefore, a data transfer packet need not be sent to the Host Interface Adapter 214.

Detailed Description Text (466):
If PRIOR.sub.-- MISS is not set, then control is followed to Step 6138 where information is added to the data transfer request packet which is sent from the Index Processor 236 to the Host Interface Adapter 204. The data transfer request packet identifies the data in Non-volatile Storage to be transferred or the area in Non-volatile Storage to which data is to be written. The data transfer request packet contains a segment information portion for each segment referenced by the file access command. In particular, the address in Non-Volatile Storage 220 where the segment to transfer is located, an address in NVS which is a pointer to the flags field in the File Descriptor 508, and the IXP identifier are loaded in the segment information portion of the data transfer request packet. After the transfer is complete, the HIA clears the SEGMENT.sub.-- BUSY flag in the File Descriptor 508 using this packet. After the HIA completes the data transfer, the FLAGS word from the data transfer packet is written to NVS by the HIA. Step 6140 increments a counter to the next segment information portion in the data transfer request packet.

Detailed Description Text (468):
Step 6148 sets the SEGMENT.sub.-- BUSY flag in the File Descriptor 508 for the segment being referenced. After the segment is marked as busy, the data transfer request packet is sent to the Host Interface Adapter 204 from which the Command Packet was sent as shown by Step 6150.

Detailed Description Text (470):
The description now returns to control Path 6142n for processing WRITE commands. The processing performed for WRITE commands is used to track the proportion of File Space 502 which is occupied by segments which have been written, and take the appropriate actions. Decision Step 6144 tests whether the segment being referenced is either nailed or belongs to a resident file. This is done by testing the NAIL flag in the File Descriptor 508. If the segment is either nailed or belongs to a Resident file, then control Path 6144y is followed to decision Step 6158. Decision Step 6158 tests whether the segment is an Orphan. An orphaned segments is segment belonging to a Resident File which was stored in Cache File Space 522. Once all the allotted segments in Resident File Space 524 have been assigned, Cache File Space is used to store segments of Resident files. If the segment is not an orphan, the processing proceeds to control Path 6142y.

Detailed Description Text (471):
If the segment is not nailed or does not belong to a Resident file, then control Path 6144n is followed to decision Step 6160, or if the segment is nailed and is an orphan, processing proceeds also proceeds to decision Step 6160. Decision Step 6160 tests whether the SEGMENT.sub.-- WRITTEN flag in the File Descriptor 508 is set. If the SEGMENT.sub.-- WRITTEN flag is already set, then the WRITTEN.sub.-- TO counters do not need to be incremented, and processing proceeds to control Path 6142y. If the segment has not yet been written, then control is followed to decision Step 6162.

Detailed Description Text (473):
Step 6166 sets the STANDBY flag in the data transfer packet which is sent to the Host Interface Adapter 204. The STANDBY flag indicates that the copy of the File Descriptor Table 506 which is kept in a second Non-Volatile Storage 220 module should be updated with the same information that is to be stored in the main File Descriptor Table. Step 6166 also updates the Backpanel ID portion of the data

transfer packet. The data transfer packet is updated with the Backpanel ID of the Backpanel having the Non-Volatile Storage 220 module in which the copy of the File Descriptor Table is stored. Both the main File Descriptor Table and the copy are stored at the same physical address within each of the Non-Volatile Storage 220 modules. Processing then proceeds to control Path 6142y.

Detailed Description Text (477):
Step 6180 sets the PRIOR.sub.-- MISS flag and also sets the appropriate bit in the SEGMENT.sub.-- MISS.sub.-- TEMPLATE. Control Path 6180p is followed to decision Step 6182. Decision Step 6182 tests whether there are more segments referenced in the Command Packet by comparing the SEG.sub.-- CTR (the number of segments processed thus far) with the segment count (SEG.sub.-- CNT) in the Command Packet. If there are segments remaining to be processed, then processing proceeds to Step 6184 where the LOOP-CODE routine is performed. LOOP-CODE increments SET.sub.-- CTR and the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET so that the next requested segment can be processed. Processing then follows control Path 6184p which returns control to Step 6122 to search and process the next requested segment.

Detailed Description Text (481):
Processing begins by invoking the SEARCH routine as indicated by Step 6220. The SEARCH routine checks whether the segment referenced in the Command Packet is present in File Space 522. If the segment is present, the HIT.sub.-- FLAG is set. Decision Step 6222 checks whether the segment was found by testing the HIT.sub.-- FLAG. If the HIT.sub.-- FLAG is set, then control Path 6222y is followed to decision Step 6224. Decision Step 6224 tests whether the state of the segment located in File Space is SEGMENT.sub.-- DISABLED. If the state of the segment is not equal to SEGMENT.sub.-- DISABLED, then processing proceeds to decision Step 6226. Step 6226 tests whether the segment stage is equal to SEGMENT.sub.-- BUSY. If the segment is not busy, then processing proceeds to decision Step 6230. Step 6230 tests whether the segment stage is equal to STAGE.sub.-- PENDING.

Detailed Description Text (483):
Step 6234 copies the DISK.sub.-- NUMBERs, DISK.sub.-- ADDRESSes, and GROUP.sub.-- ID from the Command Packet 1252 to the File Descriptor 508. Decision Step 6236 tests whether there is a standby File Descriptor Table 506. The Standby flag is set at system initialization if there is more than one Non-Volatile Storage 220 module. If there is, Step 6238 stores the information referenced in Step 6234 into the duplicate File Descriptor Table. Otherwise, control Path 6236n is followed to Step 6240.

Detailed Description Text (484):
If the Cache Sticking Power Flag (CSPF) in the Command Packet 1252 is set, then the SICKY.sub.-- COUNTER flag in the File Descriptor 508 is set at Step 6240. When the cache replacement algorithm passes over a segment whose STICKY.sub.-- COUNTER flag is set, the STICKY.sub.-- COUNTER is cleared. When the STICKY.sub.-- SLAVE counter is cleared, the associated segment is once against eligible for reassignment. Those skilled in the art will recognize that if the STICKY.sub.-- COUNTER was implemented with more than one bit, the counter could be decremented with each pass of the cache replacement algorithm. When the counter reached zero, the associated segment would be eligible for cache replacement.

Detailed Description Text (485):
Decision Step 6242 tests whether the command in the Command Packet is equal to STAGE WITHOUT DATA. If the command is not a STAGE WITHOUT DATA command, then processing continues with Step 6244. Step 6244 stores the identifier for the Host Interface Adapter 204 to which the data transfer request will be sent and the Index Processor 236 identifier for the Index Processor sending the data transfer request in the data transfer request. The identifiers are stored in the request for the purpose of routing the request in the Street 234. In addition, Step 6244 sets the SEGMENT.sub.-- BUSY flag in the File Descriptor 508, after which, processing proceeds to decision Step 6246.

Detailed Description Text (486):
Decision Step 6246 tests whether the command is STAGE SEGMENT. The extra processing associated with the STAGE BLOCKS command is shown by control Path 6246n. STAGE

SEGMENT processing is illustrated by control Path 6246y. For a STAGE SEGMENT command, step 6248 sets the Segment Valid flag in the data transfer request which is sent to the Host Interface Adapter 204. After the Host Interface Adapter has successfully stored the segment in Non-Volatile Storage, the Segment Valid Flag is written to the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor 508 by the Host Interface Adapter. The TOTAL.sub.-- SEGMENT.sub.-- VALID flag indicates that all blocks within the segment are valid. Control path 6248p is followed to decision Step 6250.

Detailed Description Text (487):
Decision Step 6250 checks whether there is a duplicate File Descriptor Table 506 by testing the Standby flag. If the answer is yes, then Step 6252 sets a standby flag in the data transfer request that is sent to the Host Interface Adapter 204. Otherwise, control proceeds directly to Step 6254.

Detailed Description Text (488):
Step 6254 stores the address in Non-Volatile Storage 220 at which the Host Interface Adapter 204 is to store the data, the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE from the File Descriptor 508, and the address in Non-Volatile Storage of the SEGMENT FLAGS in the File Descriptor. The BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE is used by the Host Interface Adapter when it processes a STAGE BLOCKS command. The address of the SEGMENT FLAGS is provided to the Host Interface Adapter so that the SEGMENT.sub.-- BUSY flag within the File Descriptor can be cleared after the Host Interface Adapter has completed the data transfer. Step 6256 stores a flag-word to the data transfer request, wherein the flag-word is stored by the Host Interface Adapter into the SEGMENT FLAGS in the File Descriptor and the SEGMENT.sub.-- BUSY flag in the File Descriptor is effectively cleared.

Detailed Description Text (493):
If the command is not STAGE SEGMENT, that is the command is a STAGE BLOCKS, Step 6246 forces control Path 6246n to decision Step 6276. Because STAGE BLOCKS is used for user data being created in the Host 10, there is no copy as yet on disk. Therefore, segments created by STAGE BLOCKS must be flagged as newly written. Step 6276 checks whether the SEGMENT.sub.-- WRITTEN flag within the File Descriptor 508 is set. If the segment has been written, then control Path 6276y is followed to 6250. Otherwise, control Path 6276n is followed to decision Step 6278. Step 6278 checks whether the segment is nailed by testing the NAIL flag in the File Descriptor. If the segment is nailed, decision Step 6280 checks whether the segment is an orphan by testing the ORPHAN flag in the File Descriptor. Otherwise, Step 6280 is skipped. If the segment is nailed and is an orphan segment, then the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is incremented at Step 6282 and processing proceeds to decision Step 6250. Note that the count of written segments is being kept only for Cache File Space 522 and not for Nail Space 523 or Resident File Space 524.

Detailed Description Text (494):
For a STAGE WITHOUT DATA command, decision Step 6242 forces control Path 6242y to Step 6284 instead of control Path 6242n. Step 6284 clears the STAGE.sub.-- PENDING flag in the File Descriptor and sets the appropriate miss flag in the Status Packet. Step 6286 clears the SEGMENT.sub.-- BUSY flag in the File Descriptor because the STAGE WITHOUT DATA command does not write data to a segment in cache. If there is a standby File Descriptor Table 506, then SEGMENT.sub.-- BUSY flag in the duplicate File Descriptor is also cleared.

Detailed Description Text (496):
The remaining discussion of the STAGE routine involves handling of exception conditions. If the HIT.sub.-- cache, then control Path 6222n is followed to Step 6292. If the segment in process is not in cache as expected, the condition may be due to the segment being assigned to a different file. The STAGE request must be resubmitted from the Host 10 and reprocessed because the entire request could not be honored. Step 6292 makes the RECOMMENDED.sub.-- ACTION in the Program Status Packet "Send Clear Pending Followed by Original Command." Processing then proceeds to Step 6294 which invokes the Back-out Busy processing.

Detailed Description Text (497):
Back-out Busy processing is illustrated in the flowchart fragment shown in FIG.

102B. Step 6296 indicates that the SEGMENT.sub.-- BUSY flags in the File Descriptors 508 associated with all the segments referenced by the Command Packet should be cleared. The ENDERR routine is then invoked at Step 6298. The ENDERR routine does not set up destage requests in the Program Status Packet 460 as does the END routine, but does return control to the COMMAND-BRANCH routine.

Detailed Description Text (498):
If decision Step 6224 finds that the segment located by the SEARCH routine has been disabled, then Step 6300 makes the RECOMMENDED.sub.-- ACTION in the Program Status Packet "Log the Condition and Return Status to User." This holds the suspect segment until an ensuing destage operation can flag the segment in the Host's master file directory. Back-out Busy processing is then performed at Step 6302.

Detailed Description Text (499):
If decision Step 6226 finds that the SEGMENT.sub.-- BUSY flag in the File Descriptor 508 is set, then Step 6228 invokes the WAIT routine to wait until the SEGMENT.sub.-- BUSY flag is cleared before continuing. SEGMENT.sub.-- BUSY is not normally a possibility in this processing, except in the case where data is being staged into a partially written segment.

Detailed Description Text (501):
Decision Step 6304 checks whether the segment has been written by testing the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor. If the segment has been written, then control Path 6304n is followed to Step 6292 to clear the pending states of those other segments referenced in the Command Packet, and clearing the SEGMENT.sub.-- BUSY flags. If the segment has not been written, then the request in the Command Packet may still be honored and control Path 6304y is followed to decision Step 6306.

Detailed Description Text (502):
Decision Step 6306 tests whether the command in the Command Packet is STAGE WITHOUT DATA. For a STAGE WITHOUT DATA command, control Path 6306y is followed to 6270. The request in the Command Packet may still be honored because the STAGE WITHOUT DATA command does not involve writing a segment to the Outboard File Cache 102. During the course of normal processing, the STAGE WITHOUT DATA command is not used. If the command is not STAGE WITHOUT DATA, then processing follows control Path 6306n to decision Step 6308. Decision step 6308 checks whether the state of the segment is equal to PURGE.sub.-- PENDING. If the PURGE.sub.-- PENDING flag in the File Descriptor 508 is not set, then control Path 6308n is followed to Step 6240. If the PURGE.sub.-- PENDING flag in the File Descriptor is set, then "Resend" is assigned to the RECOMMENDED.sub.-- ACTION in the Command Packet at Step 6310 and Back-out Busy processing is invoked at Step 6312.

Detailed Description Text (509):
Decision Step 6356 tests whether the segment is busy by testing the SEGMENT.sub.-- BUSY flag in the File Descriptor 508. If the segment is busy, then Step 6368 waits until the segment is no longer busy before allowing processing to continue. Once the segment is found not to be busy, processing continues to decision Step 6370. If decision Step 6370 finds that the state of the segment is pending, then processing follows control Path 6354n as discussed above. Otherwise, control is passed to decision Step 6372.

Detailed Description Text (510):
If a segment has not been written, no destaging is required. Decision Step 6372 checks whether the segment under examination has been written by testing the SEGMENT.sub.-- WRITTEN flag in the File Descriptor 508. If a segment for which destage is requested is found to not have been written, control follows control Path 6354n as discussed above. Otherwise, processing continues at decision Step 6374.

Detailed Description Text (517):
Once a hit is detected at decision Step 6392, decision Step 6400 checks whether the segment is busy by testing the SEGMENT.sub.-- BUSY flag in the File Descriptor 508. If the segment is busy, processing is suspended until the segment becomes available as indicated by Step 6402. When the segment is not busy, Step 6404 reads the second portion of the File Descriptor 508. It is worth noting that each File Descriptor is

stored in two different areas. The most frequently referenced portion of the File Descriptor is stored in a first area in Non-volatile Storage, and the less frequently referenced portion of the File Descriptor is stored in a second area in Non-volatile Storage. The division of the File Descriptors is done for performance purposes. The first eight words of each File Descriptor are referenced on most every File Cache operation, and the second eight words are less frequently referenced. Therefore the first eight words of each File Descriptor are stored in the first area and the second eight words of each File Descriptor are stored in the second area.

Detailed Description Text (520):
If the state of the segment is not PURGE.sub.-- PENDING, control Path 6412n directs processing to decision Step 6414. The processing steps conditioned upon Step 6414 relate to whether the segment was written since the time the DESTAGE command was initiated. If the segment has not been written, then control Path 6414n leads to decision Step 6416. The TOTAL.sub.-- SEGMENT.sub.-- VALID flag is tested to determine whether all the blocks in the segment contain data staged from Disk. If so, Step 6418 clears the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE in the File Descriptor 508 to indicate that all blocks are valid and no blocks have been written.

Detailed Description Text (521):
If the entire segment is not valid, the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE remains unchanged and processing proceeds to decision Step 6420. If the segment is not nailed as indicated by the NAIL flag in the File Descriptor 508, Step 6422 increments the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER. The LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is used to count the number of segments processed by this routine which were successfully destaged and not since written. This counter is then subtracted from the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER to keep the global counter of written segments up-to-date.

Detailed Description Text (522):
For nailed segments, decision Step tests whether the segment is an orphan by testing the ORPHAN flag in the File Descriptor 508. It is undesirable for nailed segments which are not orphans to influence the cache replacement algorithm. Therefore, the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is not incremented unless the nailed segment is also an orphan as indicated by decision Step 6424.

Detailed Description Text (523):
Step 6426 clears the DESTAGE.sub.-- PENDING and DESTAGE.sub.-- REPORTED flags in the File Descriptor 508 to indicate that the DESTAGE operation is complete. The updated File Descriptor is stored in both the main File Descriptor Table 506 and the backup File Descriptor Table at Step 6428. Once all segments indicated in the Command Packet 1664 have been processed, the ENDWT routine is invoked at Step 6430 to update the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER and return a status to the Host 10.

Detailed Description Text (524):
Returning to decision Step 6410, if the Host 10 was unsuccessful in destaging the identified segments, then the SEGMENT.sub.-- WRITTEN flag in the File Descriptor 508 must be reset and the pending flags must be cleared. Step 6432 sets the SEGMENT.sub.-- WRITTEN flag, stores the GROUP.sub.-- ID from the command packet, and clears the DESTAGE.sub.-- PENDING and PURGE.sub.-- PENDING flags in the File Descriptor. Control then follows control Path 6432p to control Path 6426p as described above.

Detailed Description Text (525):
At decision Step 6412, if the state of the segment as indicated by the File Descriptor 508 is PURGE.sub.-- PENDING, then the present DESTAGE COMPLETE command was sent to complete a PURGE command and the File Descriptor must be updated accordingly. If the Purge Type (PT) in the DESTAGE COMPLETE Command Packet 1664 is purge blocks, then decision Step 6434 directs processing to proceed along control Path 6434y to decision Step 6435. Decision Step 6435 tests whether the current segment under examination is either the first or the last segment having a block to be purged. For both the first and last segments, the PURGE-BLOCKs processing is invoked at Step 6436 to purge only the blocks identified in the Command Packet. If

there are blocks in the first or last segment which remain written after the purge is complete, then decision Step 6437 directs control to Step 6438 where the TOTAL.sub.-- SEGMENT.sub.-- VALID is cleared. Step 6436 sets the SEGMENT.sub.-- WRITTEN flag and clears the PURGE.sub.-- PENDING flag in the File Descriptor. Processing then proceeds along control Path 6432p as described above. If decision Step 6435 finds that the segment in process is neither the first nor the last segment, then processing proceeds to control Path 6442y as discussed below.

Detailed Description Text (526):
If decision Step 6434 does not detect Purge Type of purge blocks, then control Path 6434n is followed to decision Step 6439. If the Purge Type is purge segments, control is directed to control Path 6442y. If the Purge Type indicated in the DESTAGE COMPLETE Command Packet 1664 is Purge Leg 1, then the LEG1.sub.-- DISK.sub.-- NUMBER and LEG1.sub.-- DISK.sub.-- ADDRESS fields in the File Descriptor are cleared at Step 6441. Similarly, if the Purge Type is Purge Leg 2, then the LEG2.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- ADDRESS fields in the File Descriptor are cleared. Decision Step 6442 tests whether both legs in the File Descriptor were cleared. If not, then processing proceeds to Step 6439 as discussed above. This path is involved in purging the segment as soon as its copies on Leg-1 and Leg-2 have been destaged. Because one or both could not be successfully destaged at this time, the SEGMENT.sub.-- WRITTEN flag is set again.

Detailed Description Text (527):
Processing follows control Path 6442y to decision Step 6444. If the segment is not nailed as indicated by the NAIL flag in the File Descriptor 508, Step 6446 increments the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER. The LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is used to count the number of segments processed by this routine which were successfully destaged and not since written. This counter is then subtracted from the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER to keep the global counter of written segments up-to-date.

Detailed Description Text (528):
For nailed segments, decision Step tests whether the segment is an orphan by testing the ORPHAN flag in the File Descriptor 508. It is undesirable for nailed segments which are not orphans to influence the cache replacement algorithm. Therefore, the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is not incremented unless the nailed segment is also an orphan as indicated by decision Step 6448.

Detailed Description Text (531):
FIG. 105 contains a flowchart of the processing done by the Outboard File Cache for a WRITE OFF BLOCK BOUNDARY command. The processing required for a WRITE OFF BLOCK BOUNDARY command is similar to that done for READ and WRITE commands. Therefore, the same READ-WRITE routine is used with flags set to indicate that a WRITE OFF BLOCK BOUNDARY command is in process. The processing illustrated simply sets two flags which are used later in the processing of the command. Step 6472 sets the WRITE.sub.-- OFF.sub.-- BLOCK.sub.-- BOUNDARY flag which is referenced later in the READ-WRITE routine, and Step 6474 sets a WRITE.sub.-- OFF.sub.-- BLOCK.sub.-- BOUNDARY bit in the data transfer request packet which will be sent to the Host Interface Adapter 214. The READ-WRITE routine is invoked at Step 6476 to complete the remainder of processing required for the WRITE OFF BLOCK BOUNDARY command.

Detailed Description Text (559):
Step 6622 initially invokes the SEARCH processing for locating the first segment specified in the Command Packet 452. Decision Step 6624 tests whether the SEARCH processing was successful in locating the identified segment. If the segment was found, then decision Step 6626 tests whether the segment is busy by testing the SEGMENT.sub.-- BUSY flag in the File Descriptor 508. If the segment is busy, Step 6628 suspends further processing until the segment is no longer busy.

Detailed Description Text (563):
If the command in the Command Packet is a PURGE FILE command, then decision Step 6646 directs the processing to Step 6648. Step 6648 clears the LEG1.sub.-- DISK.sub.-- NUMBER and LEG1.sub.-- DISK.sub.-- ADDRESS in the File Descriptor 508 if the LG1 flag in the PURGE FILE Command Packet is set. Similarly, the LEG2.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- ADDRESS in the File Descriptor 508 if

the LG2 flag in the PURGE FILE Command Packet is set. Decision Step 6652 checks whether the Purge Type (PT) in the Command Packet is purge segment, or both legs in the File Descriptor have been cleared. If neither of the conditions in Step 6652 are found to be true, then the Purge Type is purge blocks and control is directed to decision Step 6654. Decision Step 6654 tests whether the SEGMENT.sub.-- WRITTEN flag in the File Descriptor is set. If the segment has been written, then processing proceeds to decision Step 6656 for testing whether the current segment is either the first or last segment addressed by the command. For the first and last segments, the BLOCK-PURGE processing is invoked to clear the appropriate bits in the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE in the File Descriptor. Decision Step 6660 tests whether any of the Blocks 504 in the current segment have been written by testing the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE. If there are blocks remaining in the current segment which have been written, then Step 6662 clears the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor. After Step 6662, processing proceeds along control Path 6624n. If Step 6660 finds that none of the blocks have been written, then control is directed to control Path 6652y.

Detailed Description Text (564):
Decision Step 6664 tests the SEGMENT.sub.-- WRITTEN flag in the File Descriptor 508. If the flag is set, then decision Step 6666 checks whether the NAIL flag in the File Descriptor is set. If the NAIL flag is not set, then control proceeds directly to Step 6668 where the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is incremented. The LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is later subtracted from the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER for purposes of monitoring cache usage. If the segment is nailed, decision Step 6670 tests whether the nailed segment is an orphan by checking the ORPHAN flag in the File Descriptor. For nailed segments which are not orphans, the cache usage monitoring is not affected and control is followed to Step 6672 where the PURGE processing is invoked to clean up HASH.sub.-- LINKs and clear the File Descriptor.

Detailed Description Text (565):
If decision Step 6646 finds that the command is not PURGE, then control is directed to decision Step 6674. If the SEGMENT.sub.-- WRITTEN flag is set, then Step 6676 is invoked to build a Destage Request Packet 1606. If the segment has not been written, then decision Step 6678 tests whether the Purge flag (see the DESTAGE AND PURGE FILES BY ATTRIBUTES Command Packet 1760) in the Command Packet is set. If the Purge flag is set, processing proceeds to Step 6672 as discussed above. Otherwise, control is directed to Step 6680.

Detailed Description Text (572):
If decision Step 6716 detects that the time expended processing the command has exceeded 50 milliseconds, then control is directed along Path 6716y to Step 6724. After the RECOMMENDED.sub.-- ACTION is set to Iterate, decision Step 6696 tests whether the command is RETURN SEGMENT STATE. If the test is positive, Step 6726 builds a Segment State Packet and sends it to the Host Interface Adapter 214. This is performed in addition to Step 6632 because of a limitation in the size of the transfer packet to the HIA. Step 6728 stores the segment counter in the PACKETS.sub.-- RETURNED.sub.-- COUNT field, and stores the current File Relative Segment Offset in the RESTART.sub.-- SEGMENT.sub.-- POINTER in the Status Packet 460. Decision Step 6730 tests whether the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER should be adjusted to account for segments for which the SEGMENT.sub.-- WRITTEN flag was cleared. If the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is equal to 0, then no adjustment is necessary and the END processing is invoked at Step 6732. Otherwise, the END-WRITTEN-TO processing is invoked at Step 6734.

Detailed Description Text (573):
For commands other than RETURN SEGMENT STATE, decision Step 6696 directs control to decision Step 6736. Decision Step 6736 tests whether any destage packets were built. If not, processing proceeds to Step 6728 as discussed above. If there are destage packets to process, Step 6738 is performed to build a Segment Information Packet 1662 and sent to the Host Interface Adapter 214 to store in the Status Packet 460. Step 6740 sends a data transfer request packet to the Host Interface Adapter to transfer the specified segment from Non-volatile Storage 220 to the Host 10. If decision Step 6742 finds that any of the segments for which destage is requested are disabled (SEGMENT.sub.-- DISABLED flag in the File Descriptor), then Step 6744 sets

the RECOMMENDED.sub.-- ACTION in the Status Packet 460 to Purge Disabled Segments and Resend. If no disabled segments were detected, control is directed to Step 6728 as discussed above.

Detailed Description Text (574):
FIG. 113E contains a flowchart of the processing performed for RETURN SEGMENT STATE processing of Step 6632. Where neither the SEGMENT.sub.-- WRITTEN flag nor the DESTAGE.sub.-- PENDING flag in the File Descriptor 508 are set, processing returns to control Path 6624n. If decision Step 6752 detects that either the SEGMENT.sub.-- WRITTEN or DESTAGE.sub.-- PENDING flag is set in the File Descriptor 508, then control is directed to decision Step 6754. If the segment type as specified in the SEG.sub.-- TYPE field in the RETURN SEGMENT STATE Command Packet 2106 is equal to 0, 1, or 2, then the STICKY.sub.-- COUNTER in the File Descriptor is set at Step 6756. Step 6758 builds a Segment State Packet and sends it to the HIA for sending to the Host 10 in the RETURN SEGMENT STATE Status Packet 2108. Step 6760 increments the segment counter and processing returns to control Path 6624n.

Detailed Description Text (575):
FIG. 113F illustrates a flowchart for the processing performed for MODIFY File Descriptor processing of Step 6636. If the LG1 flag in the MODIFY File Descriptor Command Packet 1772 is set, then Step 6772 replaces the LEG1.sub.-- DISK.sub.-- NUMBER and LEG1.sub.-- DISK.sub.-- ADDRESS in the File Descriptor 508 with the corresponding parameters provided in the Command Packet 1722. Similarly, if the LG2 flag in the Command Packet is set, the LEG2.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- ADDRESS are replaced with the corresponding parameters from the Command Packet. Step 6776 stores the GROUP.sub.-- ID from the Command Packet in the File Descriptor and control is returned to control Path 6624n.

Detailed Description Text (579):
On the second iteration of the main loop for a matching segment, decision Step 6810 directs control to decision Step 6822. Decision Step 6822 tests whether the SEGMENT.sub.-- BUSY flag is set in the File Descriptor 508. If it is, then Step 6824 suspends further processing until the segment is no longer busy. Once the segment is no longer busy, processing proceeds to decision Step 6826 which tests for the RETURN SEGMENT STATE COMMAND. If the command is other than RETURN SEGMENT STATE, then control is directed to decision Step 6828 where a test is made for the CLEAR PENDING command. If a CLEAR PENDING command is found, the FIX-STATE processing of Step 6830 repairs the state of the segment according to the conditions set forth in its flowchart. Control is then directed to control Path 6802n to prepare for the next iteration of the main processing loop.

Detailed Description Text (581):
Step 6836 advances the pointer to the next File Descriptor for the next iteration of the main processing loop. Decision Step 6838 tests whether the pointer has advanced beyond the end of the File Descriptor Table 506. If all segments in the Outboard File Cache 102 have not been processed, then control is directed to decision Step 6840. Decision Step 6840 tests whether the command is DESTAGE AND PURGE DISK or DESTAGE AND PURGE FILES BY ATTRIBUTES. If the command requires destaging segments and decision Step 6842 finds that the number of destage packets built is equal to the D.sub.-- CNT specified in the Command Packet 1702, then control is directed to Step 6844 where the RECOMMENDED.sub.-- ACTION is set to Iterate. If the command does not require destaging segments or if more destage packets can be built, then control is directed to decision Step 6846. Decision Step 6846 tests whether 50 milliseconds have elapsed since processing of the Command Packet began. If the allotted time has elapsed, then processing proceeds to Step 6844. Otherwise, the second pass flag is cleared at Step 6848 so that the first iteration of the main processing loop may be performed for the next File Descriptor. Control is then directed back to the beginning of the main processing loop via control path 6820p.

Detailed Description Text (586):
If decision Step 6796 detects either the DESTAGE AND PURGE FILES BY ATTRIBUTES or the PURGE FILES BY ATTRIBUTES command, the first eight words of the File Descriptor are loaded as shown by Step 6872. Decision Step 6874 checks whether the SEGMENT.sub.-- UNAVAILABLE flag in the File Descriptor 508 is set. If the flag is not set, control is directed to decision Step 6876 where the Command Packet

FILE.sub.-- IDENTIFIER is compared with the File Descriptor FILE.sub.-- IDENTIFIER after applying the ATTRIBUTES.sub.-- MASK in the Command Packet to the File Descriptor. If the values are equal, then processing proceeds to decision Step 6808 as discussed above. If the FILE.sub.-- IDENTIFIERs are not equal, control is directed to Step 6832 as discussed above.

Detailed Description Text (591):
If decision Step 6898 finds that the segment is not in a PENDING state, control is directed to decision Step 6906. Processing proceeds to Step 6908 if decision Step 6806 detects a PURGE DISK command. Step 6908 clears the LEG1.sub.-- DISK.sub.-- NUMBER in the File Descriptor 508 if it is equal to the DISK.sub.-- NUMBER specified in the PURGE DISK Command Packet 1802. Similarly, Step 6910 clears the LEG2.sub.-- DISK.sub.-- NUMBER in the File Descriptor 508 if it is equal to the DISK.sub.-- NUMBER specified in the PURGE DISK Command Packet. If both disk legs have been purged, i.e., LEG1.sub.-- DISK.sub.-- NUMBER=LEG2.sub.-- DISK.sub.-- NUMBER=0, then the PURGE processing must be performed and the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER may require adjustment. Decision Step 6914 tests whether the SEGMENT.sub.-- WRITTEN flag in the File Descriptor is set. If the segment has been written and decision Step 6916 finds that the segment is not a Nailed segment, Step 6918 increments the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER which is later subtracted from the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER. If the segment is a Nailed segment and decision Step 6920 finds that the File Descriptor indicates that the segment is an Orphan, Step 6918 increments the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER. If the Nailed segment is not an Orphan or if the segment was not written, control proceeds to Step 6922 which invokes PURGE processing to clean up HASH.sub.-- LINKs and clear the File Descriptor. Processing then continues at Step 6832 as discussed above.

Detailed Description Text (592):
If decision Step 6906 finds that the command is other than PURGE DISK and decision Step 6924 determines that the command is PURGE FILES BY ATTRIBUTES, then control is directed to decision Step 6914 as discussed above. If decision Step 6924 fails (the command is DESTAGE AND PURGE DISK or DESTAGE AND PURGE FILES BY ATTRIBUTES), processing proceeds to Step 6926 to determine whether the segment has been written. If the SEGMENT.sub.-- WRITTEN flag is set, then Step 6928 is invoked to build a Destage Request Packet 1606. If the segment has not been written, then decision Step 6930 tests whether the Purge flag (see the DESTAGE AND PURGE FILES BY ATTRIBUTES Command Packet 1760) in the Command Packet is set. If the Purge flag is set, processing proceeds to Step 6922 as discussed above. Otherwise, control is directed to Step 6832.

Detailed Description Text (594):
If decision Step 6952 finds that the entry in the Hash Table is equal to 0, then there requested segment is not in the Outboard File Cache and control is directed to Step 6954 where the Hit flag is cleared to signal a miss condition. Processing control is then returned to the point at which the SEARCH processing was invoked.

Detailed Description Text (595):
Decision Step 6952 directs control to decision Step 6956 if the Hash Table 6000 entry is non-zero. Decision Step 6956 tests whether the FILE.sub.-- IDENTIFIER in the Command Packet 452 is equal to the FILE.sub.-- IDENTIFIER in the File Descriptor 508 to which the Hash Table entry points. If the FILE.sub.-- IDENTIFIERS match, then decision Step 6958 tests whether the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSETs also match. If the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET are equal, then the requested segment has been located and Step 6960 sets the Hit flag to indicate as such. Control is then returned to the point at which the SEARCH processing was invoked.

Detailed Description Text (598):
Special processing is required for allocating Resident File Space 524, therefore, decision Step 6972 tests whether the Resident File flag (XF) in the Command Packet 1600 is set. The typical scenario is a request for access to a non-resident file, so that case will be discussed first. If the Resident File flag is not set, processing proceeds to decision Step 6974. Decision Step 6974 tests a flag which was set when a segment for which access had been requested had no blocks present in Cache File

Space 522. That is, a segment that was not a hit was previously encountered. Note that a partial miss exists when a segment exists but has some blocks missing. Processing Steps 6972 through 6980 are performed only the first time that MISS processing is performed for a group of segments specified in a Command Packet. If this is the first time that MISS has been invoked from READ-WRITE processing (decision Step 6974 fails), then SPECULATE-DECISION processing is invoked at Step 6876. SPECULATE-DECISION processing determines whether it is appropriate to speculatively reserve segments in Cache File Space 522. That is should the segments be allocated before they are explicitly referenced in a Command Packet. Decision Step 6978 tests whether the command is WRITE or WRITE OFF BLOCK BOUNDARY. The SURGE-TEST processing is invoked at Step 6980 to test whether a single file from monopolizing the available Cache File Space. Control is followed to Step 6982 to set the local prior-miss and full-miss flags. Processing then proceeds to decision Step 6984.

Detailed Description Text (600):
Decision Step 6988 tests whether the reserved segment has not been assigned to a different IXP. If the test fails, control returns to Step 6984 as discussed above. Otherwise, processing proceeds to Step 6990. If the temporary-sequential-flag was set in SPECULATE-DECISION processing, then the SEQUENTIAL.sub.-- SEGMENT flag in the File Descriptor 508 is set to indicate that the logical segment preceding the current segment is present in the Outboard File Cache 102. Step 6992 sets the ALLOCATE.sub.-- WRITE.sub.-- MISS flag in the File Descriptor if the command is either WRITE or WRITE OFF BLOCK BOUNDARY. If the Residency Required (RR) flag in the Command Packet 1600 is set, then the RESIDENT.sub.-- FILE, NAIL, ORPHAN flags in the File Descriptor are set and a temporary orphan flag is set for use in MISS-B processing. Step 6996 clears the LEG1.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- NUMBER in the File Descriptor and RELINK processing is invoked at Step 6998. RELINK processing links the File Descriptor into a hash list referenced by the Hash Table 6000. MISSB processing is invoked at Step 7000.

Detailed Description Text (602):
If the segment from Step 7004 was not a purged segment (its FILE.sub.-- IDENTIFIER is not equal to 0), then decision Step 7018 directs control to Step 7020. Step 7020 clears the lock held on the eight segment group which is under examination for cache replacement. If a lock was granted for the pointers and variables used for cache replacement, then control is directed to Step 7000. Otherwise, Step 7024 sets the NEW flag in the File Descriptor so that the segment will not be allocated during cache replacement until the REPLACEMENT CANDIDATE pointer cycles around the File Descriptor Table 506. As is made apparent in the RE-USE processing, an IXP 236 need not have obtained a lock on the cache replacement pointers and variables in order to allocate a segment. If the replacement pointers are already in use by a first IXP, a second IXP may allocate a segment which is ahead of the REPLACEMENT CANDIDATE pointer, and for this reason, the NEW bit is set to ensure that the segment allocated by the second IXP will not be reallocated until the REPLACEMENT CANDIDATE pointer cycles through the File Descriptor Table.

Detailed Description Text (603):
If decision Step 6972 finds that the Resident File (XF) flag in the Command Packet 1600 is set, then Step requests a lock used for exclusive access to the variables used in processing Resident File Space 524. Decision Step 7028 tests whether the lock was granted. If not, decision Step 7030 tests whether 50 millisecond has elapsed since processing of the current command began. When 50 milliseconds have elapsed and no lock has been granted, control is directed to decision Step 6974 as discussed above.

Detailed Description Text (605):
Step 7040 sets the NAIL and RESIDENT.sub.-- FILE flags and clears LEG1.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- NUMBER in the allocated File Descriptor. RELINK processing is invoked at Step 7042 to link the File Descriptor into the Hash Table 6000, and MISS-B processing is invoked at Step 7044.

Detailed Description Text (607):
If there are more segments to process, control is directed to decision Step 7054 which test for the WRITE and WRITE OFF BLOCK BOUNDARY commands. For the WRITE

commands, control Path 7054y is followed to decision Step 7056 where the Residency Required (RR) flag in the Command Packet 2132. If residency is required for the segments, the RECOMMENDED.sub.-- ACTION in the Status Packet 460 is set to Rescan File at Step 7058 and MISS-END processing is invoked at Step 7060. If the segment is not required to be resident and decision Step 7062 finds that the Temporary orphan flag is not set, then Step 7064 sets the RECOMMENDED.sub.-- ACTION in the Status Packet to Stage Data. Decision Step 7066 makes one final check as to whether a segment in process is referenced in a LOCK command. If the segment in process is covered by either an entry in the File Lock Descriptor Table 6502 or the Attribute Lock Descriptor Table 6504, control is directed to Step 7067 where the RECOMMENDED.sub.-- ACTION is set to Resend and Step 7068 invokes END processing. Otherwise, control is directed to Step 7060 as described above. If the Temporary orphan flag is set, then decision Step 7062 directs control to Step 7069 where the RECOMMENDED.sub.-- ACTION is set to Stage Data and Log No Resident File Space Condition. Processing then proceeds to Step 7066 as discussed above.

Detailed Description Text (609):
Decision Step 7074 tests whether either the Residency Required flag (RR) or the Resident File flag (XF) is set. If either flag is set, control is directed to Path 7054y as discussed above. Otherwise, control is directed to decision Step 7076 to test whether the total number of segments written in Cache File Space 522 is greater than 75% of the total number of segments available in Cache File Space. Once this maximum is exceeded, processing proceeds to Path 7054y as discussed above. If the maximum has not been reached, decision Step 7078 compares the Speculation Count (SC) in the Command Packet 1600 to the number of segments which have been speculatively allocated at this point in the processing. If the number of segments requested Speculation Count have been speculatively allocated, then control proceeds to Path 7054y as discussed above. Otherwise, control is directed to decision Step 7080.

Detailed Description Text (610):
If the current segment in process is not the last segment in its group of eight segments, decision Step 7080 directs processing to Step 7082 where the local prefetch mode flag is set and the count of the number of segments speculatively allocated is incremented. Step 7050 is executed as described above. If the current segment in process is the last segment in its group of eight Hash Table 6000 entries, decision Step 7084 tests whether the segment has been referenced in a LOCK command. If the segment is locked (as indicated by the File Lock Descriptor Table 6502 or the Attribute Lock Descriptor Table 6504), Step 7085 sets the RECOMMENDED.sub.-- ACTION to Resend and END processing is invoked at Step 7086.

Detailed Description Text (613):
If the Command Chain flag is not set, Step 7100 requests a lock on the pointers used in selecting one or more segments to destage. If the lock is not granted immediately, decision Step 7102 directs control to decision Step 7106. Otherwise, Step 7104 invokes DESTAGE-CHECK processing to select one or more segments to request that the Host 10 destage. Decision Step 7106 tests whether the command is WRITE or WRITE OFF BLOCK BOUNDARY. For a READ command, control is directed to Step 7098. For the WRITE commands, decision Step 7108 checks whether there are any destage requests in the Status Packet. If there are destage requests in the Status Packet, then processing proceeds to END processing. If there are no Destage Request Packets, then SURGE-TEST is invoked at Step 7110 to determine whether a file is surging and add Destage Request Packets to the Status Packet.

Detailed Description Text (614):
FIG. 119 contains a flowchart of the MISS-BA processing. Step 7114 stores the HOST.sub.-- ID and PROGRAM.sub.-- ID from the Command Packet 452 in the File Descriptor 508. Step 7116 sets the STAGE.sub.-- PENDlNG flag and stores the PATH.sub.-- ID and IXP.sub.-- NUMBER in the File Descriptor. MISS-B is invoked at Step 7118.

Detailed Description Text (615):
FIGS. 120A, 120B, 120C, and 120D contain a flowchart of FLAGS processing which tests the flags in the File Descriptor when a segment hit occurs. Step 7202 tests the SEGMENT.sub.-- BUSY flag in the File Descriptor 508 and suspends processing until the segment is no longer busy. If all the blocks in the segment have been staged or

written, decision Step 7204 directs control to decision Step 7206. If decision Steps 7206 and 7208 respectively find that the segment state is neither PURGE.sub.-- PENDING nor STAGE.sub.-- PENDING, control is directed to decision Step 7210.

Detailed Description Text (616):
If the STICKING.sub.-- MASTER flag in the File Descriptor 508 is set, Step 7212 sets the STICKING.sub.-- SLAVE flag in the File Descriptor so that the segment will not be considered for cache replacement for two round robin cycles. Decision Step 7214 directs control to Step 7216 to return control to the processing from which FLAGS was invoked if the segment in process was not allocated speculatively as indicated by the SPECULATIVE flag in the File Descriptor. Control is directed to decision Step 7218 if the segment was speculatively allocated. If the segment is speculative and is a nailed segment, control is returned to the processing from which FLAGS was invoked. Otherwise, Step 7220 clears the SPECULATIVE flag in the File Descriptor and increments the count of segments speculated in processing this command. Control is then returned as discussed above.

Detailed Description Text (620):
Special processing is required when the segment in process results in a hit and not all of the blocks of the segment have been written. Decision Step 7204 directs control to decision Step 7240 if the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor 508 is not set. If the segment in process has not been purged, its SEGMENT FLAGS will not have been cleared and control is directed to decision Step 7242. Control is directed to Step 7244 if the command is other than READ and the backpanel identifier of the backpanel in which the backup File Descriptor Table 506 is stored is provided to the HIA 214.

Detailed Description Text (621):
If the command is WRITE, decision Step 7246 fails and control is directed to decision Step 7206 as described above. Control is directed to Step 7248 for a WRITE OFF BLOCK BOUNDARY COMMAND. Step 7248 indicates to the HIA 214 that the command is WRITE OFF BLOCK BOUNDARY and processing proceeds to decision Step 7250 to test the state of the segment. If the segment state is STAGE.sub.-- PENDING, control is directed to Step 7222 as discussed above. Otherwise, decision Step 7252 tests the PURGE.sub.-- PENDING flag in the File Descriptor 508. If the segment is PURGE.sub.-- PENDING, processing proceeds to Step 7222 as described above.

Detailed Description Text (624):
Decision Step 7240 directs control to decision Step 7266 if the segment in process has been purged. If the command is a WRITE or WRITE OFF BLOCK BOUNDARY and the data to be written for the segment in process all falls a block boundary, decision Step 7266 directs control to decision Step 7268. Decision Step 7268 tests whether the Full Miss flag has been set. If the flag has not yet been set, then Step 7270 invokes SURGE-TEST processing to check whether the file to which the requested segments belong is surging, and Step 7271 sets the Full Miss flag. Processing then proceeds to Step 7258 as described above. If the write does not fall on block boundary or a full segment miss has not yet been encountered, decision Steps 7266 and 7268 respectively direct control to Step 7258.

Detailed Description Text (629):
FIG. 122 contains a flowchart of the processing performed for both FIX-STATE and FIX-STATE-1. The FIX-STATE processing clears the segment state in a File Descriptor 508 in processing a CLEAR PENDING command. Step 7332 loads the first eight words of the current File Descriptor. If the PROGRAM.sub.-- ID and HOST.sub.-- ID in the File Descriptor do not match those in the Command Packet, then decision Step 7334 directs control to return to the processing from which FIX-STATE was invoked. Otherwise, control is followed to decision Step 7336 to test whether the segment state is STAGE.sub.-- PENDING. If the state is STAGE.sub.-- PENDING, and if decision Step 7338 finds that no blocks in the segment have been written, then Step 7340 clears all flags except the NAIL and RESIDENT.sub.-- FILE flags (IS THIS BECAUSE THESE ARE USED TO DESIGNATE THAT THIS PARTICULAR SEGMENT IS PART OF RESIDENT FILE SPACE OR NAIL SPACE?). Step 7342 invokes the PURGE-SEGMENT processing to clear the other fields in the File Descriptor and control returns to the processing from which FIX-STATE was invoked.

Detailed Description Text (630):
If the segment state is not STAGE.sub.-- PENDING and instead is DESTAGE PENDING, then decision Step 7344 directs control to Step 7346 to set the SEGMENT.sub.-- WRITTEN flag because the destage operation was canceled. Step 7348 clears any other PENDING flags in the File Descriptor. If the DESTAGE.sub.-- REPORTED flag is also set, decision Step 7350 directs control to Step 7352 where the same flag is cleared. Control is then returned to the processing from which the FIX-STATE processing was invoked.

Detailed Description Text (637):
Decision Step 7374 performs a test to determine whether the REPLACEMENT CANDIDATE segment should be allocated. If none of the following flags are set: SEGMENT.sub.-- WRITTEN, SEGMENT.sub.-- BUSY, STAGE.sub.-- PENDING, PURGE.sub.-- PENDING, DESTAGE.sub.-- PENDING, BAD, or PRE-USE, the segment may be eligible for allocation and control is directed to decision Step 7376. Decision Step 7376 test whether STICKING.sub.-- SLAVE, NEW, or NAIL flags are set in the File Descriptor 508 of the REPLACEMENT CANDIDATE. If none of the flags are set, the segment remains eligible for allocation and control is directed to decision Step 7378.

Detailed Description Text (638):
If the REPLACEMENT CANDIDATE segment has not been purged (its FILE.sub.-- IDENTIFIER is not equal to 0), decision Step 7378 directs control to Step 7380 where HASH processing is invoked to determine the Hash Table 6000 entry for the REPLACEMENT CANDIDATE segment. A lock for exclusive access is requested for the group of segments identified by the group of eight Hash Table entries as indicated by Step 7382. If the lock is not granted, decision Step 7384 directs control to examine another segment for allocation. Otherwise, control is directed to Decision Step 7386 to make a final check before committing to allocation of the segment. If none of the following flags are set: SEGMENT.sub.-- WRITTEN, SEGMENT.sub.-- BUSY, STAGE.sub.-- PENDING, PURGE.sub.-- PENDING, DESTAGE.sub.-- PENDING, BAD, or PRE-USE, the segment is still eligible for allocation and control is directed to decision Step 7388. Updating of the global cache replacement pointers is conditional upon the IXP 214 having been granted the corresponding lock. Therefore, decision Step 7388 tests whether the lock was granted. If the lock was granted, the local copy of the pointers are stored back to the global pointer storage area and the lock on the pointers is cleared at Step 7390. Control is then returned to the processing from which REUSE was invoked.

Detailed Description Text (639):
If at decision Step 7386 the segment is not eligible for allocation, control is directed to decision Step 7392. If the SEGMENT.sub.-- UNAVAILABLE flag is not set, decision Step 7394 tests whether the lock on the cache replacement pointers was granted. Control returns to control Path 7362y if the lock was not granted. If the lock was granted, the NEW, STICKING.sub.-- COUNTER, and DESTAGE.sub.-- REPORTED values in the File Descriptor 508 are cleared as indicated by Steps 7396 and 7398. Control then returns to control Path 7362y to examine the next File Descriptor. Note that in an alternative embodiment, the STICKING.sub.-- COUNTER could be decremented, whereby a variable cache replacement priority level would result. In the exemplary embodiment, the STICKING.sub.-- COUNTER is a flag. This results in a segment having its STICKING.sub.-- COUNTER set not being subject to cache replacement for at least one round robin cycle. If multiple bits were used to implement the STICKING.sub.-- COUNTER, selected segments could be removed from consideration for cache replacement for multiple round robin cycles.

Detailed Description Text (640):
If the segment belongs to other than Cache File Space, the SEGMENT.sub.-- UNAVAILABLE flag will be set and decision Step 7392 directs control to Step 7400. Step 7400 advances the REPLACEMENT CANDIDATE pointer to the File Descriptor 508 referenced by the HASH.sub.-- LINK in the current File Descriptor. The UPPER BOUND and LOWER BOUND destage pointers are decremented at Step 7402 and processing continues at control Path 7362y.

Detailed Description Text (642):
When decision Step 7376 finds that either the NEW, STICKING.sub.-- COUNTER, or NAIL flags in the File Descriptor are set, control is directed to decision Step 7406 to

test whether the NAIL flag is set. Control is directed to decision Step 7410 if the segment is a nailed segment and an orphan as shown by Steps 7406 and 7408. If decision Step 7410 finds that there is no Resident File Space 524 available, control is directed to decision Step 7392 as described above. Control is directed to decision Step 7378 if there is Resident File Space available.

Detailed Description Text (646):
If decision Step 7426 finds that a lock was not granted for the cache replacement pointers, control is directed to decision Step 7374 as described above. If the lock was granted, Step 7428 reads the File Descriptor 508 referenced by the RECENTLY USED ZONE pointer. Decision Step 7430 tests whether the RECENTLY USED ZONE segment is nailed. If the segment is not nailed, and decision Step 7432 finds that the SEGMENT.sub.-- UNAVAILABLE flag is set, then Step 7434 sets the RECENTLY USED ZONE pointer to the HASH.sub.-- LINK of the current RECENTLY USED ZONE File Descriptor. Decision Step 7432 directs control to Step 7436 if the SEGMENT.sub.-- UNAVAILABLE flag is not set. The REPLACEMENT CANDIDATE and RECENTLY USED ZONE pointers are stored in global storage so that they may be read by the other IXPs 214 and control is directed to decision Step 7374 via control Path 7436p.

Detailed Description Text (649):
Step 7458 increments the count of segments reserved to the IXP 214, adds an entry to the list of segments preallocated to the IXP, temporarily saves the FILE.sub.-- IDENTIFIER from the reserved segment, clears the FILE.sub.-- IDENTIFIER field in the File Descriptor 508, sets the PRE-USE flag in the File Descriptor, stores the number of the IXP in the IXP.sub.-- NUMBER in the File Descriptor, and stores the updated File Descriptor in the primary and backup File Descriptor Tables 506.

Detailed Description Text (654):
If the SEGMENT.sub.-- UNAVAILABLE flag for the Destage Candidate segment is set, control is directed to Step 7516. Step 7516 adjusts the DESTAGE CANDIDATE, UPPER BOUND, and LOWER BOUND pointers. The DESTAGE CANDIDATE pointer is made to point to the HASH.sub.-- LINK value in the File Descriptor. In addition, the UPPER BOUND and LOWER BOUND pointers are adjusted such that the predetermined relative positions are maintained between the DESTAGE CANDIDATE, UPPER BOUND, and LOWER BOUND pointers. Control then returns to decision Step 7504 as described above. If the SEGMENT.sub.-- UNAVAILABLE flag is not set, control directed to decision Step 7518.

Detailed Description Text (655):
If the segment is a nailed segment, the NAIL flag in the File Descriptor 508 will be set and decision Step 7518 will direct control to decision Step 7520. Decision Step 7520 directs control to Step 7522 if the segment is not an orphan. Destaging of nailed segments and segments in Resident File Space is not performed during the normal course of processing. Therefore, the DESTAGE CANDIDATES, LOWER BOUND, and UPPER BOUND pointers are adjusted accordingly. Step 7522 locks the UPPER BOUND and LOWER BOUND pointers. Decision Step 7523 tests whether the segment belongs to a Resident File. Segments belonging to Resident Files have the RESIDENT.sub.-- FILE flag in their respective File Descriptors set. If the segment does not belong to a Resident File, decision Step 7523 directs control to Step 7524. Step 7524 advances the DESTAGE CANDIDATE and LOWER BOUND pointers to the first File Descriptor beyond Nail Space 523. The UPPER BOUND pointer is made to point to the File Descriptor is 32 File Descriptors beyond the LOWER BOUND pointer in the File Descriptor Table. After releasing the lock on the LOWER BOUND and UPPER BOUND pointers control is returned to decision Step 7504 as described above. Decision Step 7523 directs control to Step 7526 if the segment belongs to an Resident File. Because Resident File Space 524 resides at the end of a storage Module 732, Step 7526 advances the DESTAGE CANDIDATE LOWER BOUND, and UPPER BOUND pointers to the next storage module. Step 7524 advances the pointers beyond the Nail Space 523 which resides at the beginning of each storage module 732.

Detailed Description Text (656):
If the Destage Candidate segment is neither a nailed segment nor a segment which belongs to a Resident File, decision Step 7518 directs control to decision Step 7528. If either the STICKING.sub.-- SLAVE, DESTAGE.sub.-- REPORTED, or NEW flag is set, the segment need not be destaged and decision Step 7528 directs control to decision Step 7529. Decision Step 7529 tests whether more than 75% of the segments

in Cache File Space 522 have been written and not destaged (cache-tight condition). If the cache-tight condition is detected, control is directed to decision Step 7530 to continue the search for segments to destage. Otherwise, decision Step 7529 returns control to decision Step 7502. Decision Step 7528 direct control to decision Step 7530 if the STICKING.sub.-- SLAVE, DESTAGE.sub.-- REPORTED, and NEW flag are all cleared. If either of the segment has not been written or the segment is in any PENDING state, then the segment need not be destaged and decision Step 7530 returns control to decision Step 7502 to begin processing the next segment.

Detailed Description Text (657):
Step 7532 invokes HASH processing to obtain an index into the HASH TABLE 6000 for the Destage Candidate segment. Step 7534 requests a lock on the group of eight pointers in the Hash Table to which the Hash Table index identified at Step 7532 belongs. Decision Step 7536 tests whether the lock requested Step 7534 was immediately granted. If the lock was not granted immediately, control is return to decision Step 7502 to attempt to find another segment to destage. Decision Step 7536 directs control to Step 7538 if the lock was granted immediately. Step 7538 reads the File Descriptor referenced by Hash Table entry. Step 7540 invokes SEARCH processing to locate the File Descriptor of the Destage Candidate segment on the hash list. Step 7542 reads the second half of the File Descriptor so that the flags contained therein may be tested. Processing proceeds to decision Step 7544 to test whether the Host 10 which submitted the Command Packet 452 is capable of destaging the Destage Candidate segment.

Detailed Description Text (662):
Decision Step 7560 tests whether the DESTAGE CANDIDATE pointer is ahead of the lower bound of the Destage Zone as determined LOWER BOUND pointer. If the DESTAGE CANDIDATE pointer is not ahead of the LOWER BOUND pointer decision Step 7560 returns control to decision Step 7502 in an attempt to locate a different segment to destage. Otherwise, control is directed Step 7562. Step 7562 decrements the DESTAGE CANDIDATE pointer and processing proceeds to decision Step 7564. If 75% or fewer of the segments in Cache File Space 522 have been written and not destaged, then a cache-tight condition is not present and processing proceeds to Step 7566. Step 7566 stores the DESTAGE CANDIDATE pointer and releases the lock held on the DESTAGE CANDIDATE pointer. Control is then returned the processing from which the DESTAGE-CHECK processing was invoked. If decision Step 7564 detects a cache-tight condition, control is directed to decision Step 7568. If there are no Destage Request Packets 1606 present in the Status Packet 460, then decision Step 7568 returns to control to decision Step 7504 in an attempt to locate more segments to destage. If there were segments identified for destaging, decision Step 7568 directs control to Step 7570 where the PRIORITY.sub.-- DESTAGE flag in the Status Packet is set. Processing then proceeds to Step 7566 as described above.

Detailed Description Text (663):
FIG. 127 contains a flow chart of RELINK processing which links a File Descriptor into a hash list of File Descriptors. Step 7620 stores the PROGRAM.sub.-- ID, HOST.sub.-- ID, FILE.sub.-- IDENTIFIER, and FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET from the Command Packet 452 to the File Descriptor 508. If the command is not ALLOCATE, decision Step 7622 directs control to Step 7624. Otherwise, Step 7624 is skipped. Step 7624 clears all the flags in the File Descriptor except RESIDENT.sub.-- FILE, ALLOCATE.sub.-- WRITE.sub.-- MISS, SEQUENTIAL.sub.-- SEGMENT, and ORPHAN. Step 7622 further stores the PATH.sub.-- ID and IXP.sub.-- NUMBER and sets the STAGE.sub.-- PENDING flag in the File Descriptor.

Detailed Description Text (665):
If decision Step 7628 finds that processing is in a speculative mode, then Step 7630 sets the SPECULATIVE flag in the File Descriptor 508. Otherwise, Step 7630 is skipped and processing proceeds to Step 7632 where the File Descriptor is stored in the backup File Descriptor Table. Step 7634 links the File Descriptor to the preceding File Descriptor by storing the address of the File Descriptor in the HASH.sub.-- LINK of the preceding File Descriptor. If the File Descriptor being added is the first on the hash list, then the address of the File Descriptor is stored in the Hash Table 6000 entry. Control is then returned to the point at which the RELINK processing was invoked.

Detailed Description Text (668):
If the File Descriptor is not the first File Descriptor on the hash list, decision Step 7640 directs control to decision Step 7646. If there are no File Descriptors on the hash list which follow the File Descriptor to be removed, then control is returned as described above. Otherwise, decision Step 7646 directs control to Step 7648 which waits until the BUSY flag in the preceding File Descriptor is not set. Once the preceding File Descriptor is no longer set, Step 7650 stores the HASH.sub.-- LINK from the File Descriptor being removed in the HASH.sub.-- LINK of the preceding File Descriptor and in the backup File Descriptor Table. Control is then returned to the processing from which DELINK was invoked.

Detailed Description Text (670):
FIGS. 130A and 130B contain a flowchart of the processing for detecting whether a file is surging. A file is surging if new segments are being added to the end of file at an excessive rate. This processing also checks for a cache tight condition which is an excessive number of segments written. SURGE-TEST processing begins at Step 7792 where the Resident File (XF) flag in Command Packet 452 is tested to determine whether the file referenced in the Command Packet is a Resident File. Resident Files are not subject to the same constraints as normal files and are not limited by the SURGE-TEST processing. Thus, for Resident Files control is returned to the processing from which SURGE-TEST processing was invoked. For normal files control is directed to Step 7794 to test whether the command is WRITE OFF BLOCK BOUNDARY. SURGE-TEST processing is bypassed when a WRITE OFF BLOCK BOUNDARY command is detected because it is unlikely that a file is surging when a WRITE OFF BLOCK BOUNDARY command has been issued. Control is directed Step 7796 if a WRITE OFF BLOCK BOUNDARY command is detected.

Detailed Description Text (673):
Decision Step 7810 tests the Residency Required (RR) flag in the Command Packet 452. The Residency Required flag indicates that all segments referenced by the Command Packet must be Resident File Space 524. Therefore, control is returned to the processing from which the SURGE-TEST processing was invoked if the Residency Required flag is set If the Residency Required flag is not set control is directed decision Step 7812.

Detailed Description Text (676):
Decision Step 7826 tests whether a segment was located in by SEARCH processing and performs other tests if a segment was located. If a segment was located, as indicated by the HIT flag, and the SEGMENT.sub.-- WRITTEN flag is set, the SEGMENT.sub.-- BUSY flag is not set, DESTAGE.sub.-- PENDING flag is not set, and the PURGE.sub.-- PENDING flag is not set, then control is directed to decision Step 7828. Decision Step 7828 tests whether the processing from which the SURGE-TEST was invoked is MISSED-END. If the caller was MISS-END and the DESTAGE.sub.-- REPORTED flag in the File Descriptor is not set, decision Step 7830 directs control to Step 7832. Step 7832 clears the NEW flag in the File Descriptor. Processing proceeds to Step 7834 where DESTAGE-GROUP processing is invoked. DESTAGE-GROUP processing prepares a group of segments for the Host 10 to destage. Step 7836 clears the lock held on the group of eight Hash Table entries and sets the PRIORITY.sub.-- DESTAGE flag in the Status Packet 460. The Priority Destage flag indicates to the Host 10 that the segments referenced in the Destage Request Packet should be processed immediately. If the processing from which SURGE-TEST was invoked is MISS-END, decision Step 7838 directs that control be returned to MISS-END so that a miss status can be returned to Host. Otherwise, control is directed Step 7840 where the RECOMMENDED.sub.-- ACTION in the Status Packet is set to Destage Data and then Resend. END processing is invoked at Step 7842 to complete SURGE-TEST processing.

Detailed Description Text (683):
Decision Step 7852 tests the Speculation Count (SC) Command Packet 452. If the specified number of segments to be speculatively allocated is equal to zero, control is returned to the processing from which SPECULATE-DECISION was invoked. Otherwise, control is directed to decision Step 7854. If the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET specified in the Command Packet references the first segment in a group of eight segment as referenced by Hash Table 6000, and Segment Relative Block Offset (SRBO) is equal to zero as tested at decision Step 7856, control is directed to Step 7846 as described above. Decision Step 7854 directs control to Step

7858 if the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET is not referenced by the first pointer in a group of eight pointers in Hash Table 6000. Step 7858 obtains a new index into the Hash Table by subtracting one from the present index into the Hash Table. This is done to determine whether the segment preceding the segment referenced by FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET is present in the Cache File Space. Step 7860 invokes SEARCH processing to locate the preceding segment. If the preceding segment is present Cache File Space decision Step 7862 directs control to Step 7846 where the Temporary Sequential flag is set as described above. If the preceding segment is not present in cache, control is returned to the processing from which SPECULATE-DECISION was invoked.

Detailed Description Text (684):
FIGS. 132A and 132B contain a flowchart of the DESTAGE-GROUP processing which gathers a group of segments to be included in a Destage Request Packet 1606. DESTAGE-GROUP attempts to identify up to eight continuous segments to include in a Destage Request Packet 1606. Processing begins at Step 7940 where a local segment counter is initialized to count the number of segments identified by the Destage Request Packet and the FILE.sub.-- IDENTIFIER, LEG1.sub.-- DISK.sub.-- NUMBER, and LEG2.sub.-- DISK.sub.-- NUMBER are stored in the Status Packet 460. Processing proceeds to Step 7942 where the DESTAGE.sub.-- REPORTED flag File Descriptor 508 is set. Decision Step 7944 tests the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor to determine whether every block in the segment is valid. If the segment is not entirely valid, decision Step 7944 directs control to Step 7946. Step 7946 stores the number of segments identified for destaging in the Destage Request Packet 1606, and increments the number of Destage Request Packets included in the Destage Request Table of the Status Packet 460. Control is then returned to the processing from which the DESTAGE-GROUP processing was invoked.

Detailed Description Text (685):
If decision Step 7944 finds that the entire segment is valid, control is directed decision Step 7948 where the SEQUENTIAL.sub.-- SEGMENT flag in the File Descriptor 508 is tested. If the current segment has already been identified as a segment which is part of a contiguous group of segments, control is directed to decision Step 7950. Otherwise, control is directed to decision Step 7952 to perform an additional test. Decision Step 7952 tests whether the DESTAGE-GROUP processing was invoked as a result of a miss condition encountered in processing a WRITE command. If the present processing was invoked as a result of a write-missed condition, control is directed to decision Step 7950. Otherwise, control is directed to Step 7946 as described above.

Detailed Description Text (686):
Decision Step 7950 tests whether the present segment is the last segment in a group of eight segments as identified by the Hash Table 6000. Rather than obtaining a lock on the next group of eight segments, if the present segment is at the end of the current group of eight segments, control is directed to Step 7946 to complete the storage of information in the Destage Request Packet 1606. If the present segment is not the last segment in a group of eight segments, control is directed Step 7953. To prepare for processing the next segment, Step 7952 increments the FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET and the index into the Hash Table. Step 7954 invokes SEARCH processing to locate the next segment in the hash list. Decision Step 7956 directs control to Step 7946 if the next contiguous segment was not located in the hash list. Otherwise, control is directed decision Step 7958 to test whether the entire segment is valid and has been written. If the TOTAL.sub.-- SEGMENT.sub.-- VALID flag and the SEGMENT.sub.-- WRITTEN flag indicate that either the entire segment is not valid or the segment has not been written, processing proceeds to Step 7946 to complete DESTAGE-GROUP processing. If the entire segment is valid and has been written, control is directed to decision Step 7960.

Detailed Description Text (687):
Decision Step 7960 tests whether the segment in process has already been reported for destaging. If the segment has already been reported for destage, the DESTAGE.sub.-- REPORTED flag in the File Descriptor will have been set, and control will be directed to Step 7946 as described above. Decision Step 7960 directs control to decision Step 7962 if the present segment has not been reported for destaging. Decision Step 7962 tests whether the current segment in process is contiguous with

the prior segment. This is accomplished by comparing the LEG1.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- NUMBER of the present segment to that of the prior segment. In addition, the LEG1.sub.-- DISK.sub.-- ADDRESS and LEG2.sub.-- DISK.sub.-- ADDRESS of the present segment are also compared to that of the prior segment. For the segments to be contiguous the disk numbers of the prior segment and the present segment must be equal and the disk address of the prior segment must be exactly one less the disk address of the present segment. If the present and prior segments are contiguous, decision Step 7962 returns control to Step 7942 in an attempt to locate another contiguous segment. Otherwise, control is directed to Step 7946 to complete DESTAGE-GROUP processing as described above.

Detailed Description Text (689):
For LOGICAL-SCAN and for PHYSICAL-SCAN processing, Step 7986 sets the segment count in the Segment Information Packet 1674 equal to one. Step 7988 stores the PROGRAM.sub.-- ID and HOST.sub.-- ID in the File Descriptor 508 and in the backup File Descriptor. The SEGMENT.sub.-- BUSY flag is set and the DESTAGE.sub.-- REPORTED flag is cleared in the File Descriptor. The PATH.sub.-- ID and IXP.sub.-- # are also stored in the File Descriptor. The Host Interface Adapter is provided with the address of the data to be destaged and is instructed to clear the SEGMENT.sub.-- BUSY flag and the SEGMENT.sub.-- WRITTEN flag when it has completed transfer of the data from the Outboard File Cache to the Host 10.

Detailed Description Text (690):
Decision Step 7990 tests whether the command in the Command Packet 452 is DESTAGE. If the command is DESTAGE, control is directed to Step 7992 where the DESTAGE.sub.-- PENDING flag in the File Descriptor 508 is set. If the command is other than DESTAGE, decision Step 7990 tests whether the PURGE Flag (PF) is set. Control is directed to Step 7992 if the PURGE Flag is not set, otherwise Step 7996 sets the PURGE.sub.-- PENDING flag in the File Descriptor. Step 7998 informs the Host Interface Adapter whether a backup File Descriptor is present and increments the segment counter in the Segment Information Packet 1674.

Detailed Description Text (692):
Decision Step 8006 tests the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in File Descriptor 508. If all the Blocks 504 in the Segment 503 contain valid data, control is directed Step 8008 where the remainder of the Segment Information Packet 1674 is written with the appropriate data from the File Descriptor. If there are Blocks in the Segment which do not contain valid data, Step 8010 sets the Special Processing Require (SPR) flag and the Segment Not Valid (SNV) flag in the Segment Information Packet Processing then proceeds to Step 8008 as described above. Control is then returned to the processing from which the DESTAGE-BUILD processing was invoked.

Detailed Description Text (695):
FIG. 136 contains a flowchart of the FIX-STATE-FOR-HITS processing. The FIX-STATE-FOR-HITS processing clears the segment state in a File Descriptor 508. Decision Step 8070 tests whether the segment state is STAGE.sub.-- PENDING. If the state is STAGE.sub.-- PENDING, and if decision Step 8072 finds that no blocks in the segment have been written, then Step 8074 clears all flags except the NAIL and RESIDENT.sub.-- FILE flags which are left in their original state.

Detailed Description Text (696):
If the segment state is not STAGE.sub.-- PENDING and instead is DESTAGE PENDING, then decision Step 8076 directs control to Step 8078 to set the SEGMENT.sub.-- WRITTEN flag. Step 8080 clears any other PENDING flags in the File Descriptor. If the DESTAGE.sub.-- REPORTED flag is also set, decision Step 8082 directs control to Step 8084 where the same flag is cleared. Control is then returned to the processing from which the FIX-STATE processing was invoked.

Detailed Description Text (698):
FIG. 137 contains a flowchart of the processing performed in purging a segment from the Outboard File Cache. Before the segment is purged, Step 8092 waits until the segment is not busy as indicated by the SEGMENT.sub.-- BUSY flag in the File Descriptor 508. Once the SEGMENT.sub.-- BUSY flag is cleared, Step 8094 removes the File Descriptor from its hash list by setting the pointer which points to the File Descriptor to the HASH.sub.-- LIK in the File Descriptor being purged.

Detailed Description Text (699):
If the SEGMENT.sub.-- DISABLED flag is set, decision Step 8096 directs control to Step 8098 where the SEGMENT.sub.-- UNAVAILABLE flag is set and the HASH.sub.-- LINK in the File Descriptor 508 for the segment being purged is set to the physical address of the next File Descriptor in the File Descriptor Table 506. Control is directed to Step 8100 to clear the HASH.sub.-- LINK in the File Descriptor for the segment being purged if the SEGMENT.sub.-- DISABLED flag is not set.

Detailed Description Text (700):
Decision Step 8102 tests whether the segment being purged is an orphan segment. If it is, Step 8104 clears the RESIDENT.sub.-- FILE flag. Otherwise, control proceeds directly to Step 8106. The FILE.sub.-- IDENTIFIER, FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET, BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE, and all the flags except the NAIL, RESIDENT.sub.-- FILE, and SEGMENT.sub.-- UNAVAILABLE are cleared in the File Descriptor for the segment being purged at Step 8106. In addition, the modified File Descriptor is stored in the primary and backup File Descriptor Tables 506.

Detailed Description Text (701):
If the SEGMENT.sub.-- UNAVAILABLE flag is set, decision Step 8108 directs control to Step 8110 where control is returned to the processing from which the PURGE-SEGMENT processing was invoked. Otherwise, decision Step 8112 tests whether the segment being purged is part of Resident File Space 524 by testing the RESIDENT.sub.-- FILE flag in the File Descriptor. If it is, Step 8114 invokes GIVE-RESIDENT-FILE processing to return the purged segment to the list of free Resident File segments and control is returned to the processing from which PURGE-SEGMENT was invoked. For a segment not in Resident File Space control is directed to decision Step 8116 to test whether the segment is a nailed segment. Step 8118 returns the segment being purged to the list of free segments available to allocate to requests for nailed segments if the segment is nailed. Control is then returned to the processing from which PURGE-SEGMENT was invoked.

Detailed Description Text (705):
FIGS. 139A-E contain a flowchart of the processing for the ALLOCATE command. The ALLOCATE command preassigns segments according to the parameters specified in the Command Packet 452. Decision Step 8136 tests whether the Resident file flag (XF) in the Command Packet is set. If the command is for allocating segments in Resident File Space 524, control is directed to Step 8138 to search for the requested segment. Otherwise, processing proceeds to decision Step 8140. Decision Step 8040 directs control to decision 8142 if the Nail flag in the Command Packet is set. Decision Step 8142 tests whether there is any Nail Space 523 available. If the allotted Nail Space is all in use, control is directed Step 8144 where the RECOMMENDED.sub.-- ACTION in the Status Packet 460 is set to Remove Some Nailed Segments and then Resend. Step 8146 invokes ENDERR processing to end the processing of the command. If there is Nail Space available, decision Step 8142 directs control Step 8138.

Detailed Description Text (706):
If neither the Resident file flag (XF) nor the Nail Flag (NF) in the Command Packet 452 is set, control is directed decision Step 8148. If fewer than 75% of the segments in Cache File Space 522 have been written, Step 8148 directs control to Step 8138 to invoke SEARCH processing. Otherwise, control is directed to Step 8150 where the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER is reloaded. If the most recent value of GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER now in hand, decision Step 8152 again checks whether 75% of the segments in Cache File Space have been written. If fewer than 75% of the segments have been written, control is directed Step 8138. Decision Step 8152 directs control Step 8154 if 75% or more of the segments in Cache File Space have been written. Step 8154 sets the RECOMMENDED.sub.-- ACTION in the Status Packet 460 to Resend, and Step 8156 invokes CACHE-TIGHT processing.

Detailed Description Text (707):
Step 8138 invokes SEARCH processing to determine whether the segment in process is present in the Outboard File Cache 102. Decision Step 8158 tests whether the SEARCH

processing was successful in locating the segment. If the requested segments were not found in the Outboard File Cache decision Step 8158 directs control to Step 8160. Step 8160 sets the RECOMMENDED.sub.-- ACTION to Resend and invokes END processing has been locked by a LOCK command. Processing proceeds to decision Step 8162 to test whether the Resident file flag (XRF) in the Command Packet is set. If the command is for allocation of segments in Resident File Space 524, decision Step 8162 directs control Step 8164 where a lock is requested on the variables used for managing Resident File Space. The variables used to manage Resident File Space include: a pointer to the head of the list of available segments in Resident File Space, a pointer to the tail of the list of segments available in Resident File Space, a counter of a number of segments which are available in Resident File Space, the starting address of segments in Resident File Space, and a count of the lowest number of segments which were available in Resident File Space in the last five stays.

Detailed Description Text (708):
Decision Step 8166 tests whether the lock requested at Step 8164 was granted. If the lock has not been not granted, decision Step 8168 returns control to decision Step 8166 until 50 milliseconds have elapsed since processing of the command first began. If the lock could not be granted control is directed to Step 8170 where a temporary Orphan flag is set so that the segments can be allocated in Cache File Space 522 as an orphan segment. If the lock requested at Step 8164 is granted, control is directed to decision Step 8172 to test whether there is any Resident File Space 524 available to allocate. Step 8174 invokes GET-RESIDENT-FILE processing to reapportion File Space between Cache File Space and Resident File Space and allocate a segment in Resident File Space if decision Step 8172 finds that there is Resident File Space presently available.

Detailed Description Text (710):
Control is directed to decision Step 8186 if the segment needs to be allocated as a Nailed segment. If the Nail Flag (NF) in the Command Packet 1650 is set, decision Step 8186 directs control to Step 8188 where the GET-NAIL processing is invoked to allocate a segment in Nail Space 523. Processing then proceeds to Step 8190. Step 8190 sets a temporary Nail path flag to be used later in the processing. If a segment in Cache File Space 522 is to be allocated as a nailed segment, decision Step 8186 directs control Step 8192 while a lock is requested on the cache replacement pointers. Step 8194 invokes REUSE processing to allocate a segment in Cache File Space, and Step 8196 invokes DELINK processing to delink the segment from its hash list.

Detailed Description Text (711):
Processing proceeds to Step 8198 where the File Descriptor 508 for the allocated segment is updated with the appropriate information from the Command Packet and the appropriate flags are either set or cleared. In particular, the LEG1.sub.-- DISK.sub.-- NUMBER, LEG2.sub.-- DISK.sub.-- NUMBER, LEG1.sub.-- DISK.sub.-- ADDRESS, LEG2.sub.-- DISK.sub.-- ADDRESS, GROUP.sub.-- ID, STICKING.sub.-- POWER, and IXP number from the Command Packet are stored in the File Descriptor. The segment flags in the File Descriptor are cleared, except for the NAIL, SEGMENT.sub.-- UNAVAILABLE, and RESIDENT.sub.-- FILE flag, which are left in their present state. The ALLOCATED.sub.-- WRITE.sub.-- MISS and NEW flag in the File Descriptor are set.

Detailed Description Text (712):
Decision Step 8200 tests whether the Resident file flag (XF) in the Command Packet is set. If the segment belongs to a Resident file, decision Step 8202 tests whether the segment is an orphan segment. Step 8204 sets the ORPHAN flag in the File Descriptor 508 if the segment is an orphan, otherwise, only the NAIL and RESIDENT.sub.-- FILE flag in the File Descriptor are set. If the segment does not belong to a Resident File, decision Step 8208 tests the Nail Flag (NF) flag in the Command Packet. If the Command Packet specified a nailed segment, Step 8120 sets the NAIL flag in the File Descriptor. Otherwise, control proceeds directly to Step 8212.

Detailed Description Text (713):
Step 8212 invokes RELINK processing to link the File Descriptor into the hash list. Decision Step 8214 tests whether the temporary Nail Path flag was set. Step 8216

clears the Nail path flag if decision Step 8214 found that the flag was set. Otherwise, decision Step 8218 tests whether the FILE.sub.-- IDENTIFIER in the allocated File Descriptor is equal to zero. If the segment is not presently assigned to a file, Step 8220 clears the lock held on the segments in the Lock Table. Otherwise, control is directed Step 8222 where the lock on the group of eight Hash Table 6000 entries is cleared.

Detailed Description Text (720):
FIG. 141 contains a flowchart of NEW-BIT processing which tests whether the NEW flag in a File Descriptor should be set for the segment in process. Decision Step 8320 tests whether the RECENTLY USED ZONE pointer is ahead of or behind the REPLACEMENT CANDIDATE pointer. If the RECENTLY USED ZONE pointer is ahead of the REPLACEMENT CANDIDATE pointer, control is directed to decision Step 8322. Decision Steps 8322 and 8324 determine whether the File Descriptor 508 for the current segment is within the Recently Used Zone of segments. If the File Descriptor is outside the Recently Used Zone, control is returned to the processing from which the NEW-BIT processing was invoked. Otherwise, control is directed to Step 8326 which sets the NEW flag in the File Descriptor.

Detailed Description Text (726):
Once eight segments have been reserved for use by the IXP 236, control is directed Step 8378 to convert the first 64 segments of Cache File Space in the present module to Nail Space. Each of the 64 segments in Cache File Space is examined before converting the segments to Nail Space. If the File Descriptor 508 for a segment indicates that the SEGMENT.sub.-- WRITTEN, STAGE.sub.-- PENDING, DESTAGE.sub.-- PENDING, or PURGE.sub.-- PENDING flag is set, the data for that segment is either moved to one of the segments reserved at Step 8376 or moved to a segment obtained through normal cache replacement processing. The contents of each File Descriptor are also copied to the File Descriptor corresponding to the segment to which the data is copied. The File Descriptor for the segment to which the data is copied is also linked to the Hash Table 6000. PURGE processing is invoked to clear the contents of the File Descriptor for segments converted from Cache File Space to Nail Space when the SEGMENT.sub.-- WRITTEN flag is not set and the segment is not in a PENDING state. The File Descriptor of the segment being converted is removed from the hash list of which it is a part.

Detailed Description Text (735):
Decision Step 8428 tests the SEGMENT.sub.-- UNAVAILABLE flag in the File Descriptor and directs control around the processing for converting the segment if it is unavailable. Otherwise, control is directed to decision Step 8430. If the HOSTNAIL flag of the File Descriptor indicates that the segment has not been allocated (it was on the list of available segments in Nail Space and removed above), control is directed to Step 8432. Step 8432 clears the NAIL, HOSTNAIL, HASH.sub.-- LINK, and FILE.sub.-- IDENTIFIER in the File Descriptor. If the segment has not been allocated, control is directed to Step 8434 to convert the nailed segment which has been allocated.

Detailed Description Text (736):
Step 8434 removes the File Descriptor 508 from the head of the list of available segment in Nail Space 523. The selected segment will be the segment to which the data from the segment to be converted will be moved. Step 8436 specifies the necessary steps for converting the selected segment from Nail Space to Cache File Space 522. The File Descriptor for the segment being converted is copied to the File Descriptor selected at Step 8434. The data in the segment being converted is copied to the segment selected at Step 8434. The File Descriptor selected at Step 8434 is linked into the list of which the converted segment was a part, and the FLAGs, FILE.sub.-- IDENTIFIER, FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET, and HASH.sub.-- LINK are cleared in the segment to be converted to complete the conversion.

Detailed Description Text (741):
Decision Step 8474 directs control to Step 8484 once the lock has been granted. Step 8484 increments the total number of segments available in Nail Space 523, clears the HOSTNAIL flag in the File Descriptor, and decrements the quantity of Total Nail Space which is currently in use.

Detailed Description Text (749):
Once eight segments have been reserved for use by the IXP 236, control is directed Step 8528 to convert the last 64 segments of Cache File Space to Resident File Space. Each of the last 64 segments segment of Cache File Space is examined before converting the segments to Resident File Space. If the File Descriptor 508 for a segment indicates that the SEGMENT.sub.-- WRITTEN, STAGE.sub.-- PENDING, DESTAGE.sub.-- PENDING, or PURGE.sub.-- PENDING flag is set, the data for that segment is either moved to one of the segments reserved at Step 8526 or moved to a segment obtained through normal cache replacement processing. The contents of each File Descriptor are also copied to the File Descriptor corresponding to the segment to which the data is copied. The File Descriptor for the segment to which the data is copied is also linked to the Hash Table 6000. PURGE processing is invoked to clear the contents of the File Descriptor for segments converted from Cache File Space to Resident File Space when the SEGMENT.sub.-- WRITTEN flag is not set and the segment is not in a PENDING state. The File Descriptor of the segment being converted is removed from the hash list of which it is a part.

Detailed Description Text (757):
After the list of available segments has been scanned, the File Descriptor Table 506 is scanned for segments in Resident File Space which have been written and are to be converted. Step 8576 begins with the first storage Module 732. Step 8578 determines the range of File Descriptors 508 in the File Descriptor Table to scan for segments to convert. The File Descriptor for the first segment within a storage Module which is undergoing conversion is read at Step 8580. If the SEGMENT.sub.-- UNAVAILABLE flag in the File Descriptor is set, decision Step 8582 directs control to decision Step 8584. Decision Step 8584 tests whether the File Descriptor in process is the last in the range undergoing conversion to Cache File Space 522. Step 8586 advances to the next File Descriptor to convert if there are more to process and control returns to decision Step 8580 to read the File Descriptor.

Detailed Description Text (758):
If decision Step 8582 finds the SEGMENT.sub.-- UNAVAILABLE flag is not set and decision Step 8588 finds that the NAIL flag in the File Descriptor is not set, control is directed to decision Step 8584 as described above. Otherwise, control is directed to Step 8590 where a lock is requested on the group of eight entries in the Hash Table 6000 to which the File Descriptor in process is linked. Decision Step 8592 tests whether the lock was granted. If the lock was not granted, Step 8594 waits for eight microseconds and returns control to decision Step 8592. When the lock has been granted, control is directed to decision Step 8596.

Detailed Description Text (759):
If the segment being converted has been written, Step 8598 increments the LOCAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER so that the GLOBAL.sub.-- WRITTEN.sub.-- TO.sub.-- COUNTER may be adjusted after all the segments have been converted. Decision Step 8600 tests the SEGMENT.sub.-- BUSY flag in the File Descriptor 508, and Step 8602 waits until the SEGMENT.sub.-- BUSY flag is cleared before processing continues. Step 8604 clears the RESIDENT.sub.-- FILE and NAIL flags in the File Descriptor for the segment being converted, and Step 8606 clears the lock on the group of eight Hash Table 6000 entries.

Detailed Description Paragraph Table (1):

Word	Bit	Definition
0	0-5	Valid Flag (VF) 460a indicates whether the Program Status Packet contains valid status information. If VF=0, then the Program Status Packet does not contain valid status information. If the VF=1, then the Program Status Packet does contain valid status information.
0	6-17	Reserved as referenced by 460b.
0	18-35	UPI.sub.-- NUMBER 460c is the Universal Processor Interrupt (UPI) number associated with the Outboard File Cache interface.
1	0-3	Reserved as reference by 460d.
1	4-35	PROGRAM.sub.-- ID 460e is a value which indentifies the Command Packet (or Command Packet Chain) which is associated with the Program Status Packet. If NO.sub.-- PROGRAM in the FLAGS field is set, PROGRAM.sub.-- ID is reserved. Every Outboard File Cache program issued by a Host has an associated PROGRAM.sub.-- ID which is unique within the Host. When status is returned to the Host, PROGRAM.sub.-- ID is used to relate the status to the program

to which it applies. Note that PROGRAM.sub.-- ID applies to all commands within a single program. A status is associated with a command in a command chain by using the COMMAND.sub.-- PACKET.sub.-- ADDRESS. The portion of the File Cache Handler that builds and initiates Outboard File Cache programs generates the PROGRAM.sub.-- ID. 2 0-35 COMMAND.sub.-- PACKET.sub.-- ADDRESS 406f is a value which contains the real address of the Command Packet to which the status applies. When a chain of commands is submitted to the Outboard File Cache 102 for processing, the Command Packet Address will point to the Command Packet which caused an error. If all the Command Packets in the command chain were processed without error, then the Command Packet Address points to the last Command Packet in the command chain. 3 3-35 HARDWARE.sub.-- DEPENDENT.sub.-- STATUS-1 460g is an address within Main Storage 16 which was referenced and an error was detected. The File Cache Handler Software 208 takes the RECOMMENDED.sub.-- ACTION. 4 0-35 This word is reserved and is beyond the scope of this invention. 5 0-11 RECOMMENDED.sub.-- ACTION 460i is the processing that should be performed by the File Cache Handler Software 208 upon receiving a Program Status Packet. 5 12-23 REASON 460j indicates the condition that caused the particular status to be returned. 5 24-29 COUNT 460k is the recommended number of times that the File Cache Handler Software 208 should retry when responding to the status in the Program Status Packet. For example, if the RECOMMENDED.sub.-- ACTION returned is Resend, then the Count indicates the number of times which the File Cache Handler Software 208 should resend the Command Packet. If NO.sub.-- PROGRAM in the FLAGS field is not set and the RECOMMENDED.sub.-- ACTION does not equal "no action required", this field specifies the number of times the command specified by the Command Packet pointed to by COMMAND.sub.-- PACKET.sub.-- ADDRESS should be retried. Retries apply only to that command and not to any other commands in a command chain. All retries use the same Outboard File Cache Interface to which the original command was directed. If NO.sub.-- PROGRAM in the FLAGS field is not set and RECOMMENDED.sub.-- ACTION equals "no action required", COUNT must be equal to 0. If NO.sub.-- PROGRAM in the FLAGS field is set, this field is reserved. 5 30-35 FLAGS 4601 is a set of bits that relay ancillary information. 5 30 PRIORITY.sub.-- DESTAGE indicates whether priority destage is required. If PRIORITY.sub.-- DESTAGE is set, then the Destage Request Packets in the Destage Request Table (see the READ Status Packet) refer to segments that must be destaged as soon as possible. If NO.sub.-- PROGRAM is set or DESTAGE.sub.-- REQUEST.sub.-- PACKETS is not set, PRIORITY.sub.-- DESTAGE must equal 0. 5 31 DESTAGE.sub.-- REQUEST.sub.-- PACKETS is a flag which indicates whether the Destage Request Table exists (see the READ Status Packet). If NO.sub.-- PROGRAM is set, or the status applies to an invalid command, or the status applies to a non-I/O command, then this flag must be 0. 5 32 TERMINATED.sub.-- POLLING is a flag which indicates that a Program Initiation Queue is no longer being polled. 5 33 Reserved. 5 34 NO.sub.-- PROGRAM is a flag which indicates whether the status is associated with a Command Packet. If NO.sub.-- PROGRAM is set, then the status is not associated with a Command Packet. If TERMINATED.sub.-- POLLING is set, NO.sub.-- PROGRAM must also be set. If the Program Status Packet is returned via the Status Packet Queue, NO.sub.-- PROGRAM must equal 0. This flag is beyond the scope of this invention. 5 35 Reserved and is beyond the scope of this invention. 6 0-35 STATISTICS 460m is a set of codes which indicate how successful the Outboard File Cache 208 has been in avoiding destaging file data, speculating upon the future file access commands, and the time the Outboard File Cache 208 spent in processing the Command Packet(s). 7 0-11 RECOVERY.sub.-- TIME is used to indicate to a Host 10 that the Outboard File Cache 102 is in the process of performing a set of actions to recover from an internal fault condition. The nature of the fault recovery prohibit the Outboard File Cache from responding to any commands received from a Host. When a command is received, it is not processed by the Outboard File Cache and is returned to the sending Host with a RECOMMENDED.sub.-- ACTION equal to "Resend." RECOVERY.sub.-- TIME is only used when the NO.sub.-- PROGRAM flag is not set and the RECOMMENDED.sub.-- ACTION is Resend. The value contained in RECOVERY.sub.-- TIME provides the number of six second intervals required to complete the necessary recovery actions. 7 12-35 See Words 8-127 8-127 These words contain information which is dependent upon the particular command in the Command Packet which is associated with the Program Status Packet. Words 7-119, referenced by 460n depend upon NO.sub.-- PROGRAM and COMMAND.sub.-- CODE (see the READ Status Packet), and words 120 through 127 are reserved for future use as referenced by 460o.

Detailed Description Paragraph Table (2):

Word Bit Definition
0 0-3 These bits are reserved. 0 4-7 IXP.sub.-- # identifies the last IXP which updated this File Descriptor. This flag is useful for troubleshooting. 0 8-15 The PATH.sub.-- ID indicates the Host Interface Adapter 214 that is in the process of destaging, purging, or staging the segment. 0 16-31 SEGMENT FLAGS are used to indicate various characteristics of the Segment 503 referenced by the File Descriptor 508. The flags include the following: SEGMENT.sub.-- WRITTEN is set when the Segment has been updated via a write command since the segment was assigned. This flag is cleared when the Segment is destaged. TOTAL.sub.-- SEGMENT.sub.-- VALID is set when all blocks within a Segment are valid. A segment is valid when each block in the segment contains the most recent copy of the user's data. SEGMENT.sub.-- DISABLED identifies when a hardware error was discovered for the associated segment. SPECULATIVE/ORPHAN is a context sensitive flag. If the RESIDENT.sub.-- FILE flag is set, then this flag indicates whether the segment is an orphan segment. If the RESIDENT.sub.-- FILE flag is not set, this flag indicates whether the segment was speculatively allocated. SEGMENT.sub.-- UNAVAILABLE is used to indicate whether the segment referenced by the File Descriptor is eligible for cache replacement (reassignment). If the flag is set, then cache replacement algorithm does not consider the referenced Segment for reassignment. When this flag is set, the HASH.sub.-- LINK points to the next segment available for cache replacement SEGMENT.sub.-- BUSY is used to indicate whether a read or write operation is in progress for the referenced Segment. The flag is set when a command is decoded, and remains set until the BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE has been updated. PURGE.sub.-- PENDING is used to indicate that a PURGE command found the referenced Segment had been updated, and is presently waiting for the Segment to be destaged before purging the segment. DESTAGE.sub.-- PENDING is used to indicate that a DESTAGE command is in process. The flag is set when a DESTAGE command is decoded and cleared when the corresponding DESTAGE COMPLETE command is decoded. STAGE.sub.-- PENDING is used to indicate that a READ or WRITE command resulted in a miss condition, the Segment has been assigned, and the Segment is busy until the data has been written to the Segment. ALLOCATED.sub.-- WRITE.sub.-- MISS this flag indicates that the segment was assigned by either an ALLOCATE command or a WRITE command. SEQUENTIAL.sub.-- SEGMENT is set when multiple Segments are staged together or where the Segment immediately preceding the Segment is a Segment with the same FILE.sub.-- IDENTIFIER. The flag is used for determining which Segments should be destaged as a group. RESIDENT.sub.-- FILE indicates whether the segment belongs to a Resident File. STICKING.sub.-- MASTER indicates whether the Host 10 has specified that the Segment should have a longer lifetime in the cache than Segments whose STICKING.sub.-- MASTER flag is not set. NAIL is set when a Segment is not eligible for reassignment. The Index Processor 236 sets the NAIL flag for a segment for segments which are Nailed and segments which belong to Resident files. HOSTNAIL is set when a Segment in Nail Space has been created by the ALLOCATE command. PRE-USE is set by an IXP 236 to prevent another IXP from using the Segment. This flag indicates that an IXP has reserved the segment so that the segment is immediately available for assignment by the IXP. 1-2 FILE.sub.-- IDENTIFIER identifies the File 106 to which the Segment is assigned. 3 FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET indicates the location of the Segment relative to the first Segment in the file. 4 HASH.sub.-- LINK / BADPTR / NAIL.sub.-- LINK is the pointer to the next File Descriptor in a linked list of File Descriptors. If the SEGMENT.sub.-- UNAVAILABLE flag is set, the value in this field is used as the BADPTR, which is a pointer to the next Segment whose BAD.sub.-- OR.sub.-- UNAVAILABLE.sub.-- AREA is not set. If the NAIL flag is set, then the value in this field is used as the NAIL.sub.-- LINK which points to the next File Descriptor for a nailed Segment. 5 0-20 DATA.sub.-- POINTER is the physical address in NVS 220 where the Segment is stored. It is fixed at initialization and always points to the same segment. 5 21-27 FLAG ANNEX contains more flags which indicate characteristics of the Segment 503 referenced by the File Descriptor 508. The flags include the following: STICKING.sub.-- SLAVE is used to indicate the number of times the round robin cache replacement processing should exclude the referenced segment from consideration for replacement. DESTAGE.sub.-- REPORTED is used to ensure that the IXP does not make more than one request for the Segment to be destaged. NEW is set if the Segment is within K Segments from selection for reassignment by the cache replacement algorithm. K is equal to one-half the number of Segments available in Cache File Space 522. NOTEPAD is a flag which has multiple uses. These uses will become apparent in the detailed discussion of the IXP processing. 5 28-31 BPID is

the Back Panel Identifier associated with the NVS 220 in which the Segment is located. 6-7 BLOCKS.sub.-- WRITTEN.sub.-- TEMPLATE contains one bit for each block in the segment. If a bit is set, it indicates that at some time after the segment was last destaged, the corresponding block was updated. Bit 0 of Word 6 corresponds to Block 504-0 of a Segment 503, Bit 1 of Word 6 corresponds to Block 504-1 of Segment 503, . . . , Bit 31 of Word 6 corresponds to Block 504-31 of Segment 503, Bit 0 of Word 7 corresponds to Block 504-32 of Segment 503, . . : , and Bit 31 of Word 7 corresponds to Block 504-63 of Segment 503. 8 0-7 HOST.sub.-- ID is a value identifying the Host 10 that is in the process of destaging, purging, or staging the segment. 8 8-15 GROUP.sub.-- ID indicates the group of Hosts 10 that are able to destage the segment. In particular, the Group Identifier is the group of Hosts 10 that have direct access to the Disks 106 identified by the LEG1.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- NUMBER. The group of Hosts 10 identified by the Group Identifier is called a "destage group." There are three types of destage groups: local, shared, and global. If the Group Identifier equals 0, then the segment belongs to the global destage group; if the Group Identifier equals 1, then the segment belongs to a local destage group; and if 2 <= Group Identifier <= 255, then the segment belongs to a shared destage group. The number of local destage groups is equal to the number of Hosts 10 which are coupled to the Outboard File Cache 102. There are 255 possible local destage groups. A segment which is assigned to a local destage group can only be destaged by the Host 10 to which that local destage group is assigned. Note that if GROUP.sub.-- ID = 1, the HOST.sub.-- ID contained in the FILE.sub.-- IDENTIFIER must not equal zero and must specify a connected Host 10 that is able to destage the segment. Otherwise, an error state has occurred. There are 254 possible shared destage groups. The set of Hosts 10 contained in a shared destage group is defined by the Host 10 software. The particular Hosts 10 contained in each shared destage group is dependent upon the Hosts 10 which are coupled to the Outboard File Cache 102, the Disks 106 which are shared between the Hosts 10, and the particular files shared among the Hosts 10. 8 16-23 FILE.sub.-- SESSION is used for recovery purposes when a Host fails unexpectedly. This field is beyond the scope of this invention. 8 24-31 HOST.sub.-- SESSION is Host Session Number in which the segment was assigned to a file belonging to the Host. The Host Session Number is used for recovery purposes when a Host fails unexpectedly. This field is beyond the scope of this invention. 9 0-31 LEG1.sub.-- DISK.sub.-- NUMBER identifies the first disk on which the segment is stored. "Leg" refers to the I/O Path on which the disk resides. 10 0-31 LEG2.sub.-- DISK.sub.-- NUMBER identifies the second disk on which the segment is stored. 11 LEG1.sub.-- DISK.sub.-- ADDRESS specifies the address on the leg-1 disk at which the segment is stored. 12 LEG2.sub.-- DISK.sub.-- ADDRESS specifies the address on the leg-2 disk at which the segment is stored. 13-14 These words are unused. 15 PROGRAM.sub.-- ID identifies the Outboard File Cache program issued by a Host 10 that is in the process of destaging, purging, or staging the segment.

Detailed Description Paragraph Table (3):

Word Bit Definition

0 0-17 WORD.sub.-- COUNT 870a is the number of words that will be transferred to or from the Outboard File Cache 102. If the DDW's DAC field specifies anything other than data chain pointer, WORD.sub.-- COUNT is non- zero. If DAC 870c specifies a data chain pointer, WORD.sub.-- COUNT is ignored. 0 18 DC 870b is the Data Chain Flag. If DC=0, then data chaining is not in effect. If DC=1, then data chaining is in effect. The DC in the DDW which is contained in a Command Packet is always equal to 0. Other than the DDW contained in a Command Packet, if DAC specifies data chain pointer, the DC is ignored. 0 19-21 DAC 870c is the Data Address Control which indicates how the ADDRESS field is interpreted. DAC=0 indicates that the data address is to be incremented in performing the data transfer. The ADDRESS contains the real address of the first word of a Host Local Buffer. DAC=1 indicates that the data address is to be decremented in performing the data transfer. ADDRESS contains the real address of the last word of a Host Local Buffer. DAC=2 indicates that ADDRESS contains the real address of a Host Local Buffer consisting of a single word and that no incrementing or decrementing of the ADDRESS takes place when the word is transferred. DAC=3 indicates that the contents of ADDRESS is ignored. That is, no data transfer takes place and words in the Cache Buffer are skipped according to the number specified by WORD.sub.-- COUNT. DAC=3 does not apply to write operations. This is the Skip Data

flag referenced earlier DAC=4 indicates that the ADDRESS contains the real address of the next DDW in the data chain. This type of DDW is also called a "pointer DDW". A pointer DDW cannot point to another pointer DDW. 0 22-35 Is referenced as 870d and is reserved. 1 0-35 ADDRESS 870e indicates the real address of a word in host local memory. See the description of the DAC field for an explanation of how the ADDRESS may be interpreted.

Detailed Description Paragraph Table (4):

Word Bit Definition

0-1 DATA.sub.-- DESCRIPTOR.sub.-- WORD is described by DATA.sub.-- DESCRIPTOR.sub.-- WORD in FIG. 34. 2 NEXT.sub.-- COMMAND.sub.-- PACKET is the real address of the next Command Packet in a Command Packet Chain. 3 0-3 These bits are reserved. 3 5 CCF is the Command Chain Flag. CCF=0 indicates that command chaining is not in effect, and CCF=1 indicates that command chaining is in effect. 3 6-11 LENGTH is the number of words contained in the Command Packet starting with word 4. 3 12-23 BLOCK.sub.-- COUNT is the number of blocks to be read from the Outboard File Cache 102. 3 24-35 COMMAND.sub.-- CODE indicates the command that the Outboard File Cache 102 should execute, in this case a READ. 4-5 FILE.sub.-- IDENTIFIER indicates the file from which the data is to be read. See FIG. 45. 6 FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET is the first segment, relative to the beginning of the file, that is addressed by the command. The FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET is equal to the logical track number of the segment. 7 0-3 These bits are reserved. 7 4 FS is the Force Speculation flag which indicates whether the Outboard File Cache 102, upon detecting a miss, should attempt to speculate and allocate another segment. If FS=0, then speculation is optional, and if FS=1 then speculation is required. When either RR or XF is set, or SC equals 0, FS is ignored. 7 5 RR is the Residency Required flag which indicates whether all data referenced by the command should be in Resident File Space 524. If RR=0, then residency is not required, and if RR=1, then residency is required. If RR=1 and any of the following conditions are true, then no segments are allocated or placed in a stage pending state and the command is terminated with an appropriate status. The conditions are: a) any segment referenced by the command is not resident; or b) any block referenced by the command is not valid. 7 6 XF is the Outboard File Cache Resident File flag. See the ALLOCATE Command Packet for further explanation. If RR=1, then XF is ignored. 7 7-11 SC is the Speculation Count. If the Outboard File Cache 102 detects a miss, SC indicates the number of segments that should be speculatively allocated. If either of RR or XF is set, then SC is ignored. 7 12-23 These bits are reserved. 7 24-29 SRBO is the Segment Relative Block Offset. This is the first block, relative to the beginning of the first segment, that is referenced by to the command. 7 30-35 SEG.sub.-- CNT is the number of segments that are referenced by the command.

Detailed Description Paragraph Table (5):

Word Bit Definition

0 0-3 These bits are reserved. 0 4-9 This field is beyond the scope of this invention. 0 10-17 HOST.sub.-- ID indicates a Host 10. If H=0, HOST.sub.-- ID indicates whether the file is shared by multiple Hosts 10, or local to a single Host 10. If local to single Host, the HOST.sub.-- ID indicates to which Host 10 the file is local. If HOST.sub.-- ID=0, then the file is shared. If HOST.sub.-- ID > 0, then the file is local to the Host 10 indicated by the HOST.sub.-- ID. 0 18-35 This field is beyond the scope of this invention. 1 0-3 These bits are reserved. 1 4 H is the Hardware flag. This flag differentiates between FILE.sub.-- IDENTIFIERs generated by the File Cache Handler Software 208, and those generated by the Outboard File Cache 102. If H=0, the FILE.sub.-- IDENTIFIER was generated by File Cache Handler Software 208, and if H=1, the FILE.sub.-- IDENTIFIER was generated by the Outboard File Cache 102. 1 5-35 This field is beyond the scope of this invention.

Detailed Description Paragraph Table (6):

Word Bit Definition

0-4 See the Program Status Packet 460. 5 0-11 The valid RECOMMENDED.sub.-- ACTIONS for a READ command are: "Down File Cache Interface," "Rescan File", "Resend", "Return Status to User", "Stage Data", or "Stage Data and Log No Resident File Space Condition." 5 12-35 See the Program

Status Packet 460. 6 See the Program Status Packet 460. 7 0-17 These bits are reserved. 7 18-35 DESTAGE.sub.-- REQUEST.sub.-- PACKET.sub.-- COUNT is the number of Destage Request Packets in the Destage Request Table. If the DESTAGE.sub.-- REQUEST.sub.-- PACKETS bit in the FLAGS field is not set, this field is ignored. 8 0-17 These bits are reserved. 8 18-35 SEGMENT.sub.-- MISS.sub.-- TEMPLATE is defined as follows: If the status is associated with a READ, WRITE, OR WRITE OFF BLOCK BOUNDARY, command and RECOMMENDED.sub.-- ACTION equals either "rescan file", "allocate space", "return status to user", "stage data" "stage data and log not resident file space condition", or "stage non-speculated data", this field indicates which of the segments addressed by the command were either not resident or contain invalid data. Each bit in the template maps to one of the segments. Bit 18 corresponds to the 1st segment addressed by the command, i.e., the segment specified by the command's FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET. Bit 19 correspond to the second segment addressed by the command, and so forth up to bit 35 which corresponds to the eighteenth segment addressed by the command. If a bit is set, the corresponding segment was addressed by the command and contains invalid data. If a bit is not set, either the corresponding segment was not addressed by the command or it was resident and does not contain invalid data. If the status is associated with a READ, WRITE, or WRITE OFF BLOCK BOUNDARY command and RECOMMENDED.sub.-- ACTION does not equal either "rescan file", "allocate space", "return status to user", "stage data", "stage data and log no resident file space condition", or "stage non-speculated data", this field is ignored. If the status is not associated with a READ, WRITE, or WRITE OFF BLOCK BOUNDARY command, this field is reserved. 9 This word is reserved. 10 0-29 These bits are reserved. 10 30-35 S.sub.-- COUNT is the number of segments speculated by a READ command. This value specifies the number of segments identified in SEGMENT.sub.-- MISS.sub.-- TEMPLATE that were allocated for speculation. If none of the segments identified in SEGMENT.sub.-- MISS.sub.-- TEMPLATE were allocated for speculation, S.sub.-- COUNT must equal 0. S.sub.-- COUNT must be less than or equal to 31. If the status is associated with a READ command and RECOMMENDED.sub.-- ACTION does not equal "stage data", this field is ignored. If the Status is not associated with a READ command, this field is ignored. 11-34 Destage Request Table contains up to six Destage Request Packets 1606. If the DESTAGE.sub.-- REQUEST.sub.-- PACKETS bit in the FLAGS field is not set or DESTAGE.sub.-- REQUEST.sub.-- PACKET.sub.-- COUNT equals 0, the Destage Request Table is ignored. 35-127 These words are reserved.

Detailed Description Paragraph Table (8):

Word Bit Definition

0-1 These words are reserved. 2 See the Command Packet 452. 3 0-3 These bits are reserved. 3 4-11 See the Command Packet 452. 3 12-23 These bits are reserved. 3 24-35 See the Command Packet 452. 4-6 See the READ Command Packet 1600. 7 0-4 These bits are reserved. 7 5 NF is the Nail Flag. This flag indicates that the segments allocated by the command are to be nailed. If NF=0, then the segments are not to be nailed. If NF=1, then the segments should be nailed. 7 6 XF is the Resident File flag. This flag indicates that the segments addressed by the command belong to a Resident File. If XF=0, then the file is not a Resident File. If XF=1, then the file is a Resident File. 7 8 CSPF is the Cache Sticking Power Flag. The Cache Sticking Power Flag is used to indicate that the segment is to have a longer lifetime in the Cache File Space 522. Segments whose Cache Sticking Power Flags are not set are reassigned by the cache replacement algorithm before the segments whose Cache Sticking Power Flags are set. If CSPF=0, then the segments staged by the command do not have sticking power. IF CSPF=1, the segments staged by the command have sticking power. 7 8-15 These bits are reserved. 7 16-23 See the File Descriptor 508 for a description of the GROUP.sub.-- ID. 7 24-29 These bits are reserved. 7 30-35 See the READ Command Packet 1600. 8-11 These words contain the Leg-1 and Leg-2 disk numbers and disk addresses to assign to the File Descriptor.

Detailed Description Paragraph Table (14):

Word Bit Definition

0-1 See FILE.sub.-- IDENTIFIER 1602. 2 The FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET is the offset of the first segment described by the Segment Information Packet relative to the first segment of the file. 3 0-23 These bits are reserved. 4 24-29 The FLAGS field contains a set of

flags which provide further information about the segments defined by the packet. The set of flags include the following: SPR is the Special Processing Required Flag. If this flag is set, the single segment described by the packet requires special processing by the Host 10. If SEG.sub.-- CNT > 1, then SPR must equal zero. If SEG.sub.-- CNT = 1 and SNV = 0 = SDF, then SPR must equal zero. If SEG.sub.-- CNT = 1 and either SNV = 1 or SDF = 1, then SPR must equal 1. This flag allows host software to check one flag instead of having to check several flags for infrequent conditions. SNV is the Segment Not Valid flag. If this flag is set, one or more blocks within the single segment described by this packet contain invalid data. If SEG.sub.-- CNT > 1, then SNV must equal zero. SDF is the Segment Disabled Flag. If this flag is set, the single segment described by this packet has been disabled by the Outboard File Cache 102. The data contained in the segment may be corrupt. If SEG.sub.-- CNT > 1, then SDF must equal zero. 3 30-35 SEG.sub.-- CNT is the number of segments described by this Segment Information Packet. SEG.sub.-- CNT must he greater than zero. If SEG.sub.-- CNT > 1, then all segments described by the packet are valid, logically contiguous in the file, and physically contiguous on the disks specified by LEG1.sub.-- DISK.sub.-- NUMBER and LEG2.sub.-- DISK.sub.-- NUMBER. 4 0-3 These bits are reserved. 4 4-35 LOW.sub.-- ORDER.sub.-- WRITTEN.sub.-- TO.sub.-- TEMPLATE indicates which of the first 32 blocks of a segment (blocks 0-31) have been written. Each bit in the template maps to one of the first 32 blocks. Bit 4 of word 4 corresponds to the first block, bit 5 of word 4 corresponds to the second block, and so forth. If a bit is set, then the corresponding block has been written, otherwise, the block has not been written. If SEG.sub.-- CNT = 1, then the LOW.sub.-- ORDER.sub.-- WRITTEN.sub.-- TO.sub.-- TEMPLATE contains valid information, otherwise it is ignored. 5 0-3 These bits are reserved. 5 4-35 HIGH.sub.-- ORDER.sub.-- WRITTEN.sub.-- TO.sub.-- TEMPLATE indicates which of the second 32 blocks of a segment (blocks 32-63) have been written. Bit 4 of word 5 corresponds to the thirty-third block, bit 5 of word 5 corresponds to the thirty-fourth block, and so forth. If a bit is set, then the corresponding block has been written, otherwise, the block has not been written. If SEG.sub.-- CNT = 1, then the HIGH.sub.-- ORDER.sub.-- WRITTEN.sub.-- TO.sub.-- TEMPLATE contains valid information, otherwise it is ignored. 6-9 See the READ Command Packet 1600.

Detailed Description Paragraph Table (15):

Word Bit Definition

0-1 These words are reserved. 2 See the READ Command Packet 1600. 3 0-11 See the READ Command Packet 1600. 3 12-17 FSRBO is the First Segment Relative Block Offset. This is the number of the first block, relative to the segment referenced by FILE.sub.-- RELATIVE.sub.-- SEGMENT.sub.-- OFFSET, that is to be purged. If PT does not specify purge blocks or ND is set, the FSRBO is ignored. 3 18-23 LSRBO is the Last Segment Relative Block Offset. This is the number of the last block, relative to the segment referenced by FILE.sub.-- RELATIVE.sub.-- BLOCK.sub.-- OFFSET + SEG.sub.-- CNT - 1 (the last segment referenced by the command), that is to be purged. If PT does not specify purge blocks or ND is set, the LSRBO is ignored. 3 24-35 See the READ Command Packet 1600. 4-6 See the READ Command Packet 1600. 7 0-3 These bits are reserved. 7 4-5 PT is the Purge Type. If data or control information is to be purged, that is, all segments addressed by the command are in a purge pending state, PT indicates the type of purge to be performed. Host software determines whether it is purging part of a segment or the entire segment and further determines whether leg repair is taking place on a duplexed file If PT=0, then the type of purge is purge segments. If PT=1, then the type of purge is purge blocks. If PT=2, then the type of purge is purge leg 1. If PT=3, then the type of purge is purge leg 2. If ND=1, then PT is ignored. 7 6 ND is the Not Destaged flag. This flag indicates whether or not the segments addressed by the command were successfully destaged. If the segments addressed by the command were not successfully destaged, they are marked as written and the state of the segments is changed to AVAILABLE. If ND=0, then the segments were successfully destaged. If ND=1, then the segments were not successfully destaged. 7 7-15 These bits are reserved. 7 16-23 See the (GROUP.sub.-- ID 508g in the File Descriptor 508 for a description of this field. 7 24-29 These bits are reserved. 7 30-35 See the READ Command Packet 1600. 8 This word is reserved. 9 0-3 These bits are reserved. 9 4-35 PROGRAM.sub.-- IDENTIFIER identifies the program that left the segments in a DESTAGE PENDING state. This field references the particular program (or command chain) which initiated the destage operation. It is generated from the virtual

address of the operating system structure that describes the program.

Detailed Description Paragraph Table (17):

Word Bit Definition

0-3 See the READ Command Packet 1600. 4 0-3 These bits are reserved. 4 4-35 DISK.sub.-- NUMBER is the disk identifier used by the operating system. 5-7 These words are reserved. 8 CURRENT.sub.-- SEGMENT.sub.-- POINTER is used by the Outboard File Cache 102 to locate the first segment that is to be processed by the command. CURRENT.sub.-- SEGMENT.sub.-- POINTER is 0 when the Command Packet is first sent to the Outboard File Cache. If the RECOMMENDED.sub.-- ACTION in the Status Packet is "Iterate," the CURRENT.sub.-- SEGMENT.sub.-- POINTER is assigned the RESTART.sub.-- SEGMENT.sub.-- POINTER from the Status Packet. 9 0-4 These bits are reserved. 9 5 P is the Purge flag. If the purge flag is set, segments matching the specified DISK.sub.-- NUMBER that are not destaged are purged. Segments that are destaged are subsequently purged by the DESTAGE COMPLETE command. If the purge flag is not set, then no segments are purged. 9 6-11 D.sub.-- CNT is the maximum number of segments that can be destaged. D.sub.-- CNT must be greater than one and less than or equal to 8. D.sub.-- CNT provides a mechanism that allows Host software to control an minimize the amount of buffer space that must be reserved for transferring cache segments to the Host. 9 12-35 These bits are reserved.

Detailed Description Paragraph Table (21):

Word Bit Definition

0-3 See the READ Command Packet 1600. 4-5 ATTRIBUTES.sub.-- MASK is a value that is logically ANDed with the FILE.sub.-- ID in a File Descriptor 508. 6-7 ATTRIBUTES.sub.-- ID is a value that is compared with the result of ANDing the ATTRIBUTES.sub.-- MASK with a FILE.sub.-- ID in a File Descriptor. If the ATTRIBUTES.sub.-- ID matches the result of the AND, then the segment is a candidate for the operation specified in the Command Packet. 8 See the CLEAR PENDING Command Packet 1254. 9 0-3 These bits are reserved. 9 4 B is the Bypass flag. This flag indicates that segments that cannot be destaged are to be bypassed. Examples of segments that cannot be destaged include, but are not limited to, segments in a PENDING state or segments that have the directory recovery in progress flag set. If B=0, then the segments are not bypassed; if B=1, the segments are bypassed. 9 5-35 See the DESTAGE AND PURGE DISK Command Packet 1702.

Detailed Description Paragraph Table (27):

Word Bit Definition

0-1 These words are reserved. 2 See the Command Packet 452. 3 0-3 These bits are reserved. 3 4-11 See the Command Packet 452. 3 12-23 These bits are reserved. 3 24-35 See the Command Packet 452. 4-6 See the READ Command Packet 1600. 7 0-4 These bits are reserved. 7 5 RC is the Recovery Complete flag. This flag is beyond the scope of this invention. 7 6 LG1 is the Leg-1 flag. If LG1=1, then the disk numbers and addresses for LEG1 in the File Descriptor should be modified as indicated by the Command Packet, unless RC=1. 7 7 LG2 is the Leg-2 flag. If LG2=1, then the disk numbers and addressed for LEG2 in the File Descriptor should be modified as indicated by the Command Packet, unless RC=1. 7 8-35 See the DESTAGE COMPLETE Command Packet 1664. 8-11 These words contain the Leg-1 and Leg-2 disk numbers and disk addresses to assign to the File Descriptor.

Detailed Description Paragraph Table (33):

Word Bit Definition

0-3 See the READ Command Packet 1600. 4-8 See the DESTAGE AND PURGE FILES BY ATTRIBUTES Command Packet 1760. 9 0-3 These bits are reserved. 9 4-11 HOST.sub.-- ID identifies the Host 10 for which recovery is complete. If the Purge Non-Recovered Local (PNRL) segments flag and the Local Recovery Complete (LRC) flag equal zero, then the HOST.sub.-- ID is ignored. 9 12-30 These bits are reserved. 9 31 PNRL is the Purge Non-Recovered Local segments flag. This flag indicates whether any segment within the File Space 502 for which the following conditions are true is to be purged: (1) the directory recovery in progress flag is set; and (2) the HOST.sub.-- ID in the Command Packet is equal to

the host identifier portion of the FILE.sub.-- IDENTIFIER in the File Descriptor 508; and (3) the FILE.sub.-- IDENTIFIER in the File Descriptor matches the attributes specified in the Command Packet. If PNRL = 0, then non-recovered local segments are not purged, and if PNRL = 1, then non-recovered local segments are purged. 9 32 PNRS is the Purge Non-Recovered Shared segments flag. This flag indicates whether any segment in File Space 502 for which the following conditions are true is to be purged: (1) the directory recovery in progress flag in the File Descriptor is set; and (2) the HOST.sub.-- ID in the Command Packet matches the host identifier portion of the FILE.sub.-- IDENTIFIER in the File Descriptor; and (3) the FILE.sub.-- IDENTIFIER in the File Descriptor matches the attributes specified in the Command Packet. If PNRS = 0, then the non-recovered shared segments are not purged, and if PNRS = 1, the non-recovered shared segments are purged, 9 33 LRC is the Local Recovery Complete flag. This flag indicates whether the process required to recover the local directory specified by HOST.sub.-- ID has been completed. If LRC = 0, local recovery is not completed, and if LRC = 1, the local recovery is complete. Note, fields and conditions relating to recovery of segments are beyond the scope of this invention. 9 34 SRC is the Shared Recovery Complete flag. This flag indicates whether the process required to recover the shared directory has been completed. If SRC = 0, shared recovery is not complete, and if SRC = 1, shared recovery is complete. 9 35 CP is the Clear Pendings flag. This flag indicates whether any segment in File Space 502 for which the following conditions are true is to be removed from a PENDING state: (1) the segment's state is DESTAGE PENDING, PURGE PENDING, or STAGE PENDING; and (2) the HOST.sub.-- RECOVERY.sub.-- IN.sub.-- PROGRESS flag which corresponds to the Host identified by the HOST.sub.-- ID is set; and (3) the segment was placed in a PENDING state before the most recent RECOVERY HOST command that satisfies the following conditions was processed by the Outboard File Cache 102: (a) the RECOVER HOST command did not have its probe flag set; and (b) the RECOVER HOST command had its Recover Host And Local Directory flag set; and (c) the HOST.sub.-- ID in the RECOVER HOST command is equal to the host identifier portion of the FILE.sub.-- IDENTIFIER in the File Descriptor 508. If CP = 0, the PENDING states are not cleared, and if CP = 1, the PENDING states are cleared.

Detailed Description Paragraph Table (35):

Word Bit Definition

0-1 These words are reserved. 2 See Command Packet 452. 3 0-4 These bits are reserved. 3 5-11 See Command Packet 452. 3 12-17 These bits are reserved. 3 18-23 SEG.sub.-- TYPE is the type of segments for which Segment State Packets 2110 are to be returned. If SEG.sub.-- TYPE = 0, then Segment State Packets are returned for all segments which have their SEGMENT.sub.-- WRITTEN flag set and which match the FILE.sub.-- IDENTIFIER and SEGMENT.sub.-- OFFSET parameters in the Command Packet. If SEG.sub.-- TYPE = 1, then Segment State Packets are returned for all segments which match the FILE.sub.-- IDENTIFIER and SEGMENT.sub.-- OFFSET parameters in the Command Packet. If SEG.sub.-- TYPE = 2, then one Segment State Packet is returned for the first segment whose SEGMENT.sub.-- WRITTEN flag is set and which matches the FILE.sub.-- IDENTIFIER and SEGMENT.sub.-- OFFSET parameters in the Command Packet. If SEG.sub.-- TYPE = 3, then one Segment State Packet is returned for the first segment which matches the FILE.sub.-- IDENTIFIER and SEGMENT.sub.-- OFFSET parameters in the Command Packet. If SEG.sub.-- TYPE = 4, then one Segment State Packet is returned for the first segment found in the Outboard File Cache whose SEGMENT.sub.-- WRITTEN flag is set, and which matches the DISK.sub.-- NUMBER parameter in the Command Packet If SEG.sub.-- TYPE = 5, then one Segment State Packet is returned for the first segment found in the Outboard File Cache which matches the DISK.sub.-- NUMBER parameter in the Command Packet. If SEG.sub.-- TYPE = 6, then one Segment State Packet is returned for the segment which matches the FILE.sub.-- IDENTIFIER and CURRENT.sub.-- SEGMENT.sub.-- POINTER parameters in the Command Packet. 3 24-35 See Command Packet 452. 4-6 See the READ Command Packet 1600. 7-8 See the CLEAR PENDING Command Packet 1254. 9 0-3 These bits are reserved. 9 4-35 DISK.sub.-- NUMBER is the disk identifier used by the operating system to identify the Disk 106 on which the segment is stored. DISK.sub.-- NUMBER is ignored by the Outboard File Cache unless SEG.sub.-- TYPE = 4 or SEG.sub.-- TYPE = 5.

Detailed Description Paragraph Table (37):

Word Bit Definition

0-1 See File Identifier 1602. 2 See the READ Command Packet 1600. 3 0-3 These bits are reserved. 3 4-11 HOST.sub.-- ID identifies the Host 10 whose operations put the identified segment in a PENDING state. Unless the SEGMENT.sub.-- STATE in the File Descriptor 508 is STAGE PENDING, DESTAGE PENDING, or PURGE PENDING, the contents of this field is undefined. 3 12-23 PATH.sub.-- ID is an value, internal to the Outboard File Cache 102, that identifies the physical path through which data was transferred to or from the segment. 3 24-29 STATE is the state of the segment as indicated in the File Descriptor 508. STATE = 0 indicates that the segment is AVAILABLE. STATE = 1 indicates that the state is STAGE PENDING. STATE = 2 indicates that the state is DESTAGE PENDING. STATE = 3 is reserved. STATE = 4 indicates that the state is PURGE PENDING. 3 30-35 SVF is the Segment Valid Flag. If SVF = 0, then the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor 508 is not set. If SVF = 1, then the TOTAL.sub.-- SEGMENT.sub.-- VALID flag in the File Descriptor 508 is set 4-5 See the Segment Information Packet 1662. 6 0-3 These bits are reserved. 6 4-35 PROGRAM.sub.-- IDENTIFIER identifies the program which most recently caused the identified segment to be placed in a PENDING state: Unless the STATE indicates STAGE PENDING, DESTAGE PENDING, or PURGE PENDING, the content of this field is ignored. 7 0-32 These bits are reserved. 7 33 DR is the Directory Recovery in progress flag. if DR = 1, then the segment belongs to a directory that is currently in the process of being recovered. Otherwise DR = 0. Note, recovery flags are beyond the scope of this invention. 7 34 RS is the Resident file Space flag. If RS = 1, then the segment is in Resident File Space 524. Otherwise RS = 0. 7 35 DF is the Disabled Flag. If DF = 1, the SEGMENT.sub.-- DISABLED flag in the File Descriptor was found to be set. Otherwise, DF = 0. 8-9 These words are reserved.

Detailed Description Paragraph Table (48):

Word Bit Definition

0-1 See the Data Descriptor Word 870. 2 See the Command Packet 452. 3 0-11 See the Command Packet 452. 3 12-23 BLOCK.sub.-- COUNT is the number of blocks to be written to the Outboard File Cache 102. 3 24-35 See the Command Packet 452. 4-6 See the READ Command Packet 1600. 7 0-3 These bits are reserved. 7 4 FF is the Force Fill flag. If a miss condition is encounter- ed and FF is set, bypass the sequential write check. 7 5-6 See the READ Command Packet 1600. 7 7-23 These bits are reserved. 24-35 See the READ Command Packet 1600.

Current US Cross Reference Classification (1):
707/200